Memoria

Proyecto de Fin de Carrera (PFC)

Extracción y análisis de información del servicio de red social *Twitter*, a través de la plataforma de "cloud computing" *Google App Engine*

Madrid, Mayo de 2011
Universidad Carlos III Madrid
Guillermo Cabañas Sánchez
Ingeniería Superior en Informática

Índice de Contenidos

Índice de Figuras

Índice de Tablas

Índice de Gráficos

Índice de Capturas

Datos Administrativos

Título Completo del Proyecto de Fin de Carrera (PFC):

Extracción y análisis de información del servicio de red social Twitter, a través de la plataforma de "cloud computing" Google App Engine

Autor:

Guillermo Cabañas Sánchez

Tutor:

Juan Manuel Tirado Martín

Co-director:

Daniel Higuero Alonso-Mardones

Titulación:

Ingeniero Superior en Informática

Facultad / Escuela:

Escuela Politécnica Superior

Universidad:

Universidad Carlos III de Madrid

1. Introducción

En este primer apartado se procede a enunciar las razones que han motivado el desarrollo del presente proyecto, haciendo especial hincapié en la relevancia y notable actualidad de los conceptos implicados. A continuación, se presenta la estructura de la memoria y se detallan los aspectos tratados en cada uno de los apartados que la conforman.

Como último punto de la introducción son enumeradas y descritas las definiciones de los términos, las abreviaturas y los acrónimos empleados a lo largo de todo el documento.

1.1. Motivación

En el desarrollo del proyecto *Extracción y análisis de información del servicio de red social Twitter, a través de la plataforma de "cloud computing" Google App Engine* están implicados dos de los temas más populares y candentes en la actualidad dentro del marco de las nuevas tecnologías, como son los servicios de redes sociales y el paradigma "cloud computing". Ambos son conceptos relativamente recientes, objeto de millones de referencias diarias en Internet[1][2][3] procedentes de todo el mundo, y al mismo tiempo valedores de un potencial desarrollo evidente.

Cada vez más gente considera los servicios de redes sociales como una herramienta básica en sus relaciones sociales, y cada vez son más los servicios alojados en la nube que se prestan a los usuarios de aplicaciones software. Sin lugar a dudas, son dos de las materias que han experimentado un mayor grado de avance en el mundo tecnológico durante los últimos años, y todo hace pensar que esta tendencia se mantendrá o incluso se verá reforzará en el futuro más inmediato. Con todo ello, no resulta sorprendente el elevado número de estudios (tómese como ejemplos[1][2][3]) que los

incluyen entre los actuales "Top Technology Trends" o mayores tendencias tecnológicas de nuestros días.

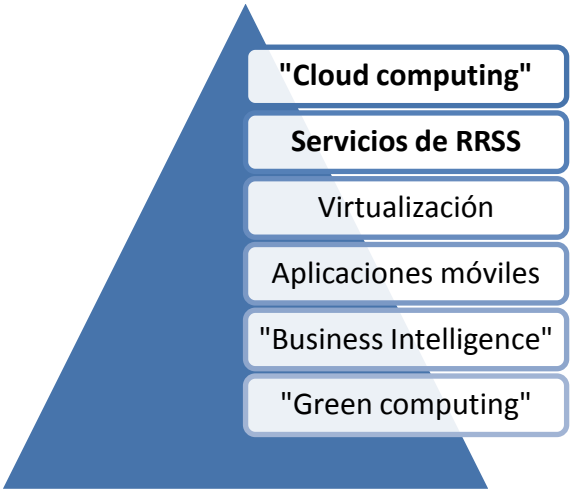


Figura 1: "Top Technology Trends" en la actualidad

La importancia de los citados conceptos ha llegado a tal extremo, que ambos aparecen concurrentemente en el día a día de una persona moderadamente relacionada con el mundo tecnológico, durante la realización de muchas de sus tareas cotidianas: consultar el correo electrónico, actualizar su perfil en una red social, ejecutar aplicaciones online, compartir fotos en la red, etc. Sin darnos cuenta, los servicios de redes sociales y el paradigma "cloud computing" se han introducido en nuestras vidas, y nos han llevado incluso a modificar ciertos hábitos, como el modo en el que gestionamos nuestras relaciones y amistades, o el lugar en el ejecutamos o alojamos nuestras aplicaciones software. Por ende y aún a riesgo de parecer exagerado, algunas de nuestras acciones diarias han dejado de ser concebibles sin su existencia.

Ciertamente los servicios de redes sociales y el paradigma "cloud computing" han generado, como muchas otras innovaciones tecnológicas, nuevas necesidades entre los usuarios que antes no existían. Por mencionar un ejemplo, hace algo menos de un lustro nadie consideraba a los servicios de redes sociales un elemento básico para la interacción con sus amistades, pero hoy en día puede decirse que sí que lo son

para un cada vez mayor porcentaje de la población. Especialmente las nuevas generaciones están asumiendo todas estas necesidades derivadas de su uso y por consiguiente, su repercusión en la sociedad aumenta fuertemente día a día. Como conclusión, puede decirse que tanto los servicios de redes sociales como el paradigma "cloud computing" suponen en estos momentos auténticas realidades, y representan necesidades para los usuarios que requieren ser cubiertas.

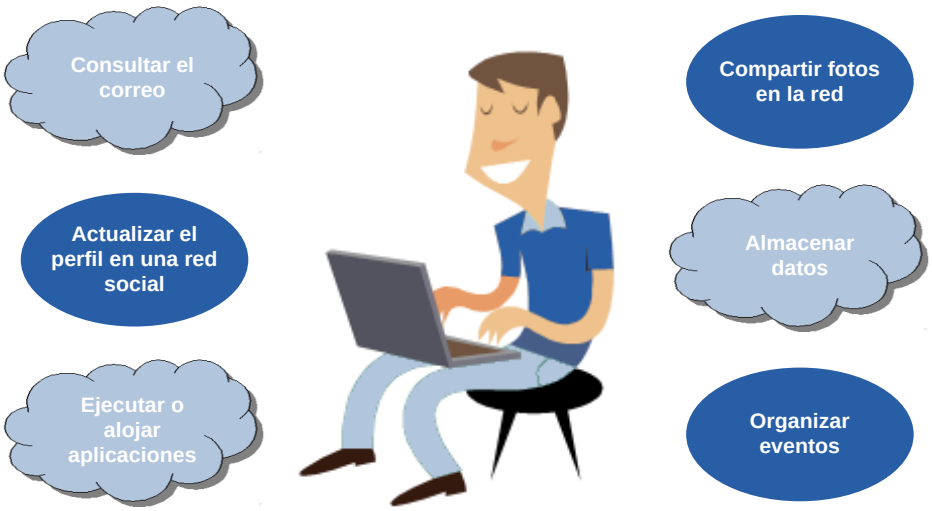


Figura 2: Los servicios de redes sociales y el paradigma "cloud computing"

En definitiva, nos encontramos ante dos conceptos que, dentro de su alcance, marcarán el devenir de la tecnología y de ciertos aspectos de nuestras vidas durante los próximos años, y por lo tanto representan un magnífico marco para la realización del proyecto de fin de carrera. El proyecto concreto que nos ocupa, tiene por objeto llevar a cabo la extracción de información del servicio de red social *Twitter*[4] mediante la plataforma *Google App Engine*[5], para posteriormente realizar su análisis e interpretación. Por su parte, los análisis elaborados se basan en distintas métricas que permiten extraer conclusiones acerca de la composición y disposición de los grafos sociales de *Twitter*, y del comportamiento de sus usuarios.

Por consiguiente, el proyecto se presenta con dos objetivos claramente diferenciados: realizar la extracción de datos a gran escala a través de una plataforma

"cloud computing", y verificar si ciertos principios que se cumplen en las relaciones personales de las sociedades corrientes, tienen su correspondencia en las sociedades paralelas que proliferan a partir de los servicios de redes sociales.

Dentro del amplio abanico disponible de servicios de redes sociales y plataformas "cloud computing" que existen en la actualidad, se han tomado dos de ellos para acotar el espectro del estudio. Los dos siguientes subapartados se dedican a detallar y argumentar la elección de *Twitter* y *Google App Engine*.

1.1.1. ¿Por qué Twitter?

A pesar de su corta edad, la importancia que tienen hoy en día los servicios de redes sociales es un tema que ya nadie discute. Su relevancia se ha hecho tan grande que ha traspasado las fronteras del campo de las nuevas tecnologías, para erigirse como una herramienta muy importante en el mundo de la comunicación, las relaciones sociales, la política, etc. Un ejemplo patente de su relevancia en la sociedad actual puede apreciarse en la Tabla 1. En ella se muestran las 10 páginas de Internet más visitadas[6] en todo el mundo, entre las que se encuentran dos dedicadas a los servicios de redes sociales: *Facebook* en la segunda posición y *Twitter* en la novena. Únicamente las páginas de los buscadores de Internet, poseen un nivel de visitas superior.

Pos.	Página Web	Propósito
1º	*Google*[7]	Buscador
2º	***Facebook***[9]	**Servicio RRSS**
3º	*Youtube*[11]	Videos
4º	*Yahoo!*[13]	Buscador
5º	*Windows Live*[14]	Buscador

Pos.	Página Web	Propósito
6º	*Baidu.com*[8]	Buscador
7º	*Wikipedia*[10]	Enciclopedia
8º	*Blogger.com*[12]	Blogs
9º	***Twitter***[4]	**Servicio RRSS**
10º	*MSN*[15]	Portal

Tabla 1: Ranking 10 páginas web más visitadas en el mundo[6]

Infinidad de servicios de redes sociales con distinta temática han aparecido durante los últimos años, algunos de los cuales se han expandido meteóricamente. En concreto, dentro del subapartado 2.1.1 se realiza una síntesis de su historia desde sus inicios en torno al año 2000 hasta la actualidad, y se dan a conocer los más populares en todo el mundo. Para el proyecto que nos ocupa, se llevó a cabo un estudio de varios de ellos con el propósito de escoger el más adecuado en el contexto dado. Durante la selección del servicio de red social objeto del análisis, los principales factores tenidos en cuenta fueron: su popularidad en términos generales, la existencia de una API abierta que permitiera hacer uso de sus recursos de forma libre, y la utilidad para el análisis del tipo de información que albergaba.

Twitter, como otros servicios de red social del tipo de *Facebook* o *MySpace*, goza de una gran popularidad. Tal y como muestra la Tabla 1, es una de las diez páginas web más visitadas en el mundo y su número de usuarios está en constante crecimiento, muy cerca de sobrepasar la barrera de los 200 millones[16]. Asimismo, *Twitter* dispone de una API pública y abierta para todo tipo de desarrolladores, descrita en el subapartado 2.1.3, que permite acceder a su información de forma práctica. Además dicha información es extremadamente densa en contenido útil dada su naturaleza "micro-blogging", aunque su análisis es complejo por tratarse de lenguaje natural. Todas estas características convierten a *Twitter* en la opción más adecuada, de entre todas las presentes en el mercado, para el propósito final del proyecto: extracción de información útil por métodos sencillos, para un posterior análisis productivo.

1.1.2. ¿Por qué Google App Engine?

Dada la relevancia que ha adquirido el paradigma "cloud computing" en los últimos años y el prometedor futuro que se le presupone por delante, son muchas las empresas y organizaciones que se han posicionado o intentan hacerlo sobre las demás ofreciendo este tipo de servicios. Existen infinidad de estudios o referencias (tómese como ejemplos[17][18][19]) de reconocidas autoridades en el mundo de las nuevas

tecnologías y de los negocios, que predicen un potente mercado entorno a dicho paradigma. El mundo empresarial es cada día más consciente de este hecho, y por ello cada vez son más las soluciones[20] disponibles para los usuarios.

En el contexto del proyecto que nos ocupa, se estudiaron las principales plataformas "cloud computing" de tipo público. Entre ellas, destacan *Google App Engine* (*GAE*)[5], *Amazon EC2*[21] y *Windows Azure*[22], que proveen aplicaciones comunes en línea accesibles desde un navegador web, mientras el software y los datos se almacenan en los servidores. Todas ellas se basan en el mismo paradigma, pese a que cada una posee sus particularidades y en algunos casos existen diferencias notables entre ellas. La Tabla 2 muestra las principales ventajas y desventajas de cada una de las tres plataformas:

	Ventajas	**Desventajas**
Google App Engine (GAE)	Generosa cuota gratuita, fácilmente escalable, abundante documentación disponible	Exclusivo para los lenguajes *Java* y *Python*
Amazon EC2	Soporta cualquier lenguaje porque se utiliza a nivel de SO y no a nivel de aplicación	Sin cuota gratuita*, difícilmente escalable, configuración compleja
Windows Azure	Varios lenguajes soportados, fácil integración empresarial, entorno de depuración	Cuota gratuita limitada, lento proceso de modificaciones y despliegue
* NOTA: con posterioridad al análisis, *Amazon EC2* habilitó una cuota gratuita limitada		

Tabla 2: Comparativa entre las principales plataformas "cloud computing"

Tras un estudio comparativo en detalle de todas ellas, finalmente se optó por *Google App Engine (GAE)* en lugar de *Amazon EC2* y *Windows Azure*, principalmente por un único motivo: su generosa cuota gratuita. Por tratarse de un proyecto universitario de fin carrera, al que no es posible asociar costes ajenos a las horas de trabajo del autor, su tutor y su co-director, el uso de cuotas gratuitas representaba un requisito

indispensable. Es por ello por lo que se descartaron *Amazon EC2*, que no disponía de cuotas gratuitas en el momento del estudio a pesar de que estas se han habilitado con posterioridad, y *Windows Azure*, cuyos recursos gratuitos[23] resultaban insuficientes para el propósito del proyecto .

1.2. Estructura de la Memoria

La memoria ha sido estructurada en apartados y subapartados atendiendo a un índice de contenidos que puede consultarse a partir de la segunda página. Asimismo, para facilitar al lector la búsqueda de figuras, tablas, gráficos y capturas a lo largo del documento, se ha incluido un índice para cada uno de los citados elementos.

En el primer apartado se procede presentar las razones que han motivado el desarrollo del proyecto, para más tarde enumerar y describir las definiciones de los términos, las abreviaturas y los acrónimos empleados. Durante el segundo apartado, se hace referencia a los principales campos del conocimiento tratados como las redes sociales, el paradigma “cloud computing” y la teoría de grafos.

En el tercer apartado de la memoria se realiza una descripción del esquema general de la arquitectura del sistema, para más tarde detallar los distintos entornos de ejecución con sus aplicaciones correspondientes. Las plataformas, lenguajes y herramientas utilizadas en las distintas fases de desarrollo, monitorización y análisis del sistema son descritas en el cuarto apartado.

Durante el apartado número cinco, se mencionan las limitaciones de la fase de extracción de la información del servicio de red social *Twitter,* atendiendo a su naturaleza, y se procede a realizar una descripción pormenorizada de cada una de las fases que componen el proceso completo de extracción, seguimiento y análisis. En el sexto apartado se exponen los resultados obtenidos del análisis de la información, y se enuncian una serie de conclusiones extraídas a partir de ellos.

La planificación del proyecto se especifica en el séptimo apartado, incluyendo tanto el calendario de ejecución como el presupuesto del mismo. A continuación, se presentan las conclusiones generales y personales del autor tras la realización del proyecto de fin de carrera.

En el apartado noveno se desarrolla la idea de posibles trabajos futuros a partir del proyecto actual, mediante el empleo de recursos no gratuitos de *GAE* o el estudio de otros servicios de redes sociales. Por último los apartados 10, 11, 12 se dedican a la sección de agradecimientos, la bibliografía y referencias, y el apéndice de la memoria.

1.3. Definiciones, Abreviaturas y Acrónimos

En los siguientes subapartados son enumeradas y descritas las definiciones de los términos, las abreviaturas y los acrónimos empleados durante de toda la memoria.

1.3.1. Definiciones

API	Conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.
Aplicación Móvil	Software desarrollado para "smartphones" o teléfonos móviles con características similares a las de un ordenador personal.
BigTable	Potente, distribuido y propietario sistema de almacenamiento de *Google*, especialmente diseñado para aplicaciones concurrentes y escalables.
Binding	Adaptación de tipo proxy o "façade" para poder utilizar una biblioteca escrita en un lenguaje nativo mediante una API.
Business Intelligence	Conjunto de estrategias y herramientas enfocadas a la administración y creación de conocimiento mediante el análisis de datos existentes en una organización o empresa.
Búsqueda en Anchura	Algoritmo para recorrer o buscar elementos en un grafo. Intuitivamente, se comienza eligiendo un nodo como elemento raíz y se exploran todos sus vecinos. A continuación para cada uno de los vecinos se exploran sus respectivos vecinos adyacentes, y así sucesivamente.

Cloud Computing	En español computación en nube, es un paradigma que permite ofrecer servicios de computación a través de Internet. La nube es una metáfora de Internet.
Elasticidad	Concepto económico que mide el grado de relación entre dos variables (p.ej. oferta y demanda, consumo de recursos y costes asociados, etc.).
Escalabilidad	Capacidad deseable de un sistema informático de cambiar su tamaño o configuración para adaptarse a circunstancias cambiantes.
Façade	Patrón de diseño que sirve para proveer de una interfaz unificada sencilla que haga de intermediaria entre un cliente y una interfaz o grupo de interfaces más complejas.
Geolocation	Identificación de la localización geográfica en el mundo real de un ordenador conectado a Internet.
Green Computing	Uso eficiente de los recursos computacionales minimizando el impacto ambiental, maximizando su viabilidad económica y asegurando deberes sociales.
Grid Computing	Paradigma de computación distribuida que permite utilizar de forma coordinada recursos heterogéneos que no estén sujetos a un control central.
Googlebot	Robot de búsqueda usado por *Google* que colecciona documentos desde la web para construir una base de datos para el motor de búsqueda *Google*.
Micro-blogging	Servicio que permite a sus usuarios enviar y publicar mensajes breves, generalmente de sólo texto. Las opciones para el envío de los mensajes varían desde sitios web, a través de SMS, mensajería instantánea o aplicaciones “ad hoc”
Pagerank	Valor numérico que representa la importancia que una página web tiene en Internet, en base a un algoritmo desarrollado por *Google* y empleado por su motor de búsqueda.
Plugin	Aplicación que se relaciona con otra para aportarle una funcionalidad nueva y generalmente muy especifica. Esta aplicación adicional es ejecutada por la aplicación principal e interactúan por medio de una API.
Podcasting	Distribución de archivos multimedia mediante un sistema de redifusión.
Power Law	Relación matemática entre dos cantidades, la variable aleatoria y su frecuencia, según la cual la frecuencia decrece en función de un exponente cuando la variable aleatoria aumenta.

Proxy	Programa o dispositivo que realiza una acción en representación de otro. En el contexto estricto de los patrones de diseño, se trata de un objeto intermediario para acceder a otro, permitiendo controlar el acceso a él
Red Social	Estructura social que se puede representar en forma de uno o varios grafos en los cuales los nodos representan individuos, a veces denominados actores, y las aristas relaciones entre ellos.
Servicio de Red Social	Aplicación web que permite gestionar la participación por parte de un usuario en una red social.
Servlet	Objeto que se ejecuta en un servidor o contenedor *JEE*, especialmente diseñado para ofrecer contenido dinámico desde un servidor web, generalmente HTML.
Streaming	Técnica de distribución de contenido (p. ej. audio o video) a través de Internet.
Virtualización	Abstracción de los recursos de una computadora que crea una capa entre el hardware de la máquina física y el sistema operativo de la máquina virtual.
Web 2.0	Segunda generación en la historia del desarrollo de tecnología Web basada en comunidades de usuarios y una gama especial de servicios, como las redes sociales, los blogs o los wikis, que fomentan la colaboración y el intercambio ágil y eficaz de información entre los usuarios de una comunidad o red social.
Whitelist	Registro de entidades que por alguna razón han sido provistas de un determinado privilegio, servicio, movilidad, acceso o reconocimiento.

Tabla 3: Definiciones del documento

1.3.2. Abreviaturas

Aprox.	Aproximadamente
Ent.	Entrantes
Etc.	Etcétera
Extract.	Extractor
GB	Gigabyte
ID	Identificador
Inc.	Incorporation

Incl.	Incluido
MB	Megabyte
Min.	Minuto
Per.	Periodo
Pos.	Posición
RT	Retweet
Últ.	Último
Unds	Unidades
Visit.	Visitas

Tabla 4: Abreviaturas del documento

1.3.3. Acrónimos

AOL	American Online
API	Application Programming Interface
BBDD	Base de Datos
BI	Business Intelligence
BSD	Berkeley Software Distribution
CDF	Cumulative Distribution Function
CPD	Centro de Procesamiento de Datos
CPU	Central Processing Unit
GAE	Google App Engine
GPL	General Public License
GQL	Google Query Language
HTTP	HyperText Transfer Protocol
I+D+i	Investigación, Desarrollo e Innovación
IaaS	Infrastructure As A Service
IDE	Integrated Development Kit
IP	Internet Protocol
IVA	Impuesto sobre el Valor Añadido

JDO	Java Data Objects
JEE	Java Platform, Enterprise Edition
JPA	Java Persistence API
JRE SE	Java Platform, Runtime Standard Edition
JSP	Java Server Pages
LGPL	Lesser General Public License
MPL	Mozilla Public License
ORM	Object Role Modeling
OS	Operative System
PaaS	Platform As A Service
PFC	Proyecto de Fin de Carrera
RDBM	Relational Data Base Modeling
RRSS	Red Social
REST	Representational State Transfer
S.A.	Sociedad Anónima
SaaS	Software As A Service
SMS	Short Message Service
SQL	Structured Query Language
UC3M	Universidad Carlos III de Madrid
UML	Unified Modeling Language
URL	Uniform Resource Locator
VAT	Value Added Tax

Tabla 5: Acrónimos del documento

2. Estado del Arte

Durante este apartado se hace referencia a los principales campos del conocimiento citados en el documento, como son las redes sociales, el paradigma "cloud computing" y la teoría de grafos. Cada campo se desglosa en una serie de subapartados, en los que se describen en más profundidad conceptos relacionados con alguna de las fases del proyecto.

2.1. Redes Sociales

Se entiende por red social un sistema abierto y en construcción permanente que involucra a conjuntos con necesidades y problemáticas comunes, que se organizan para potenciar sus recursos. De forma simplificada se trata de una estructura social que puede representarse en forma de uno o varios grafos, en los cuales los nodos representan individuos, a veces denominados actores, y las aristas las relaciones entre ellos. Desde un punto de vista genérico, las redes sociales son inherentes a la condición de ser vivo, puesto que existen desde el momento en el que éstos establecen relaciones entre ellos. Es por esto por lo que los ejemplos de redes sociales en el mundo son innumerables y variopintos: desde una simple manada de mamíferos, hasta un complejo servicio de red social web, pasando por un gremio de profesionales, por ejemplo.

De su definición puede extraerse que una red social almacena una gran cantidad de información altamente aprovechable por múltiples motivos y para diversas causas. Por ejemplo, la información relativa a los gustos de los usuarios de un popular servicio de red social web, puede ser especialmente útil para las grandes marcas de consumo en general, o centrándonos en un ejemplo más cercano, una red social puede ser la llave para encontrar a nuestros viejos compañeros de instituto o de universidad.

Sin embargo, todo este proceso de aprovechamiento de la información almacenada, requiere en la mayoría de los casos herramientas adecuadas y algoritmos complejos.

Los siguientes subapartados abordan el concepto general de servicio de red social, para centrarse a continuación en *Twitter* y su API, que han permitido extraer la información necesaria para componer los grafos sociales posteriormente analizados.

2.1.1. Servicios de Redes Sociales

Durante la última década, las sociedades avanzadas tecnológicamente han experimentado la aparición de una serie de novedosas formas de comunicación y de relación entre sus miembros, motivadas principalmente por el gran desarrollo de la red de redes, Internet. Las características de dicha red han permitido, por ejemplo, que factores como la localización geográfica hayan dejado de ser determinantes a la hora de establecer relaciones sociales entre individuos, o que los canales clásicos de comunicación (cara a cara, teléfono, etc.) hayan ido perdiendo peso en favor de otros alternativos.

Uno de los mayores exponentes de dicha tendencia es el nacimiento y elevadísima propagación de los servicios de redes sociales. En el sentido práctico, los servicios de redes sociales son aplicaciones que permiten gestionar la participación por parte de un usuario en una red social. Más concretamente, ofrecen la posibilidad de crear un perfil público o semi-público dentro de un sistema limitado, articular una lista de otros usuarios con los que se tiene una relación, y consultar o recorrer la lista de las relaciones propias y ajenas con objeto de crear otras nuevas. Ciertos servicios de redes sociales también brindan la posibilidad a sus usuarios de compartir contenido, interactuar, o crear comunidades sobre intereses o aficiones similares.

Desde sus inicios en torno al año 2000 los servicios de redes sociales han atraído a millones de usuarios, muchos de los cuales acceden a ellos de forma diaria y coti-

diana. A día de hoy existen numerosos servicios de redes sociales cubriendo un amplio espectro de intereses y prácticas, o creados alrededor de una determinada temática: música, fotos, etc. Los diferentes servicios varían tanto en el propósito con el que han sido concebidos, como en el modo en el que incorporan la nueva información, o el uso que hacen de las herramientas comunicativas (conectividad móvil, compartición de fotos/videos, etc.). La mayoría de ellos tienen un carácter general, pero existen algunos específicos para intereses concretos o actividades particulares, como por ejemplo lo es *LinkedIn*[24] para los negocios, o *Last.fm*[25] para el intercambio de gustos musicales.

Infinidad de servicios de redes sociales han aparecido durante los últimos años, algunos de los cuales se han expandido meteóricamente, mientras que otros han desaparecido. Un caso claro de crecimiento espectacular[26] es el de la red *Facebook*, con sus ya más de 350 millones de usuarios en todo el mundo a fecha de Noviembre de 2009. En el Gráfico 1 puede apreciarse como *Facebook* ha conseguido aumentar su número de usuarios de 5 a 350 millones en poco más de tres años, con un crecimiento exponencial durante el año 2009.

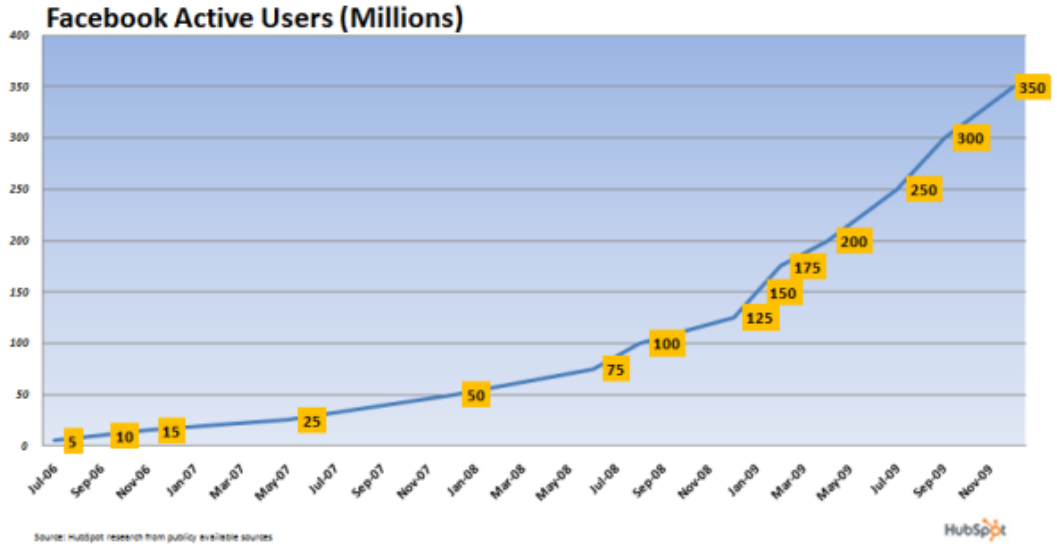


Gráfico 1: Crecimiento del número de usuarios de Facebook[27]

La expansión de cada servicio de red social no ha sido uniforme por todas las áreas geográficas del mundo, sino que su grado de popularidad puede variar ampliamente de un lugar a otro. Muchos de ellos incluso, han focalizado su mercado de usuarios en áreas concretas o sectores de población determinados, prestando una menor o inexistente atención a los otros. En términos generales, puede decirse que los servicios de redes sociales más extendidos a nivel mundial son: *Facebook*, *MySpace* y *Twitter*. En la Tabla 6 puede apreciarse un ranking con los 10 servicios de redes sociales más populares en el mundo a fecha de Febrero de 2010, ordenados por número de visitas. La información ha sido extraída de la fuente de conocimiento empresarial *Ebizmba*[28].

Pos.	Servicio RRSS	Links Ent.	Visit. Últ. Mes	Tipo	Mercado
1º	*Facebook*[9]	722.434.829	122.220.617	General	Mundial
2º	*MySpace*[29]	345.130.806	55.599.585	General	Mundial
3º	*Twitter*[4]	628.750.806	23.579.044	Micro-blog	Mundial
4º	*LinkedIn*[24]	29.370.378	11.228.746	Negocios	Mundial
5º	*Classmates*[30]	997.666	14.649.224	Estudios	EEUU-Canadá
6º	*Ning*[31]	13.032.000	5.881.943	General	Mundial
7º	*Bebo*[32]	14.368.423	3.120.062	General	Anglosajón
8º	*Hi5*[33]	8.491.287	2.176.014	General	Mundial
9º	*Tagged*[34]	399.111	3.731.972	General	Mundial
10º	*MyYearbook*[35]	921.983	3.025.772	General	EEUU

Tabla 6: Ranking 10 servicios de redes sociales más populares en el mundo[28]

A pesar de la vigencia de los datos mostrados, ha de tenerse en cuenta de que se trata de un mercado muy dinámico en constante variación. Prueba de ello es el hecho de que *Twitter*, con tan solo algo más de tres años de vida, haya sido capaz de alcanzar la tercera posición de la clasificación mundial y con previsiones de seguir aumentando en popularidad. Como apunte particular para el mercado español puede

decirse que, además de los servicios de redes sociales ya mencionados, destaca el autóctono *Tuenti*[36] nacido en enero de 2006 y con un número de usuarios que asciende a los 7 millones. Actualmente *Tuenti* es un servicio público, pero que nació como servicio privado con invitación. Éste carácter de exclusividad fue determinante durante sus inicios para lograr atraer a un gran número de personas, y en estos momentos se mantiene como el servicio de red social mayoritario en España.

2.1.2. Twitter

Twitter[4] es un servicio de red social gratuito basado en el "micro-blogging", que permite a sus usuarios enviar y leer mensajes de texto conocidos como "tweets". Dichos mensajes están compuestos por hasta 140 caracteres que aparecen en la página de perfil del autor y son notificados a sus suscriptores, también conocidos como seguidores. Por defecto el perfil y los mensajes de un usuario son públicos para todos los demás, pero existe la posibilidad de modificar el nivel de privacidad y restringir el acceso al círculo de amigos. Los usuarios pueden enviar y recibir "tweets" a través del sitio web de *Twitter*, servicio de mensajes cortos (SMS) o aplicaciones externas.

El servicio de red social nació en 2006 tras una "brainstorming" o tormenta de ideas llevada a cabo por los miembros de la junta de *Odeo*, empresa de "podcasting" que se encontraba en un bache creativo. El nombre del producto original era *Twttr*, inspirado en *Flickr*[37] y el hecho de que los códigos para los SMS en EEUU son de 5 dígitos. El primer prototipo de *Twitter* fue utilizado como un servicio interno para los empleados *Odeo*, siendo más tarde lanzado al público general en julio de 2006. Sin embargo, el verdadero punto de inflexión de la popularidad de *Twitter* ocurrió en 2007 durante la convención *South by Southwest*, cuando su uso aumento de 20.000 a 60.000 "tweets" diarios.

Actualmente *Twitter.com* está considerado como uno de los sitios web más populares en todo el mundo, como puede comprobarse en el ranking que elabora sema-

nalmente el portal *Alexa*[6] y como certifica el valor de *pagerank* 9 que le otorga el buscador *Google*[38]. Aunque las estimaciones sobre su número de usuarios diarios varían porque la compañía no facilita el número exacto de cuentas en activo, *Twitter* está reconocido como el tercer servicio de red social[28] más utilizado en el mundo, en base a su enorme número de visitantes únicos mensuales y visitas mensuales totales. Además es el que está experimentando un mayor crecimiento en el número de usuarios, con periodos de crecimiento[39] mensual que han llegado a alcanzar el 1.444%. En cuanto al tiempo medio que dedica cada usuario al servicio, también se ha visto incrementado durante los últimos años de forma notable. En el siguiente gráfico puede observarse como éste tiempo ya había superado los 17 minutos diarios de media durante el mes de Mayo de 2009.

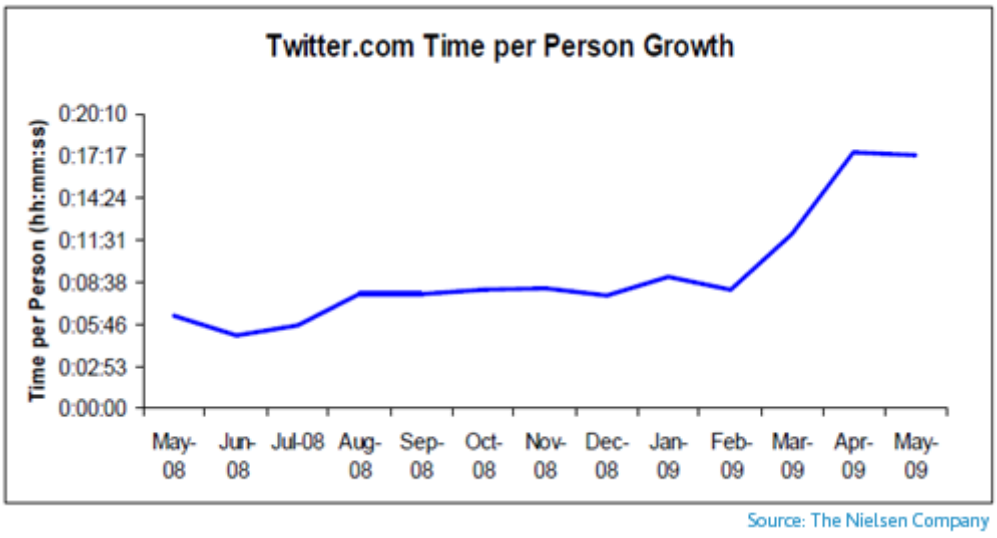


Gráfico 2: Crecimiento del tiempo diario dedicado a Twitter por sus usuarios[39]

La dimensión de *Twitter* está llegando a tal punto que para muchas personas ha dejado de ser un servicio de red social, para convertirse en una red de información pura. Prácticamente cualquier acontecimiento ocurrido en el mundo con una importancia mediana o superior tiene su reflejo en *Twitter*, y esto puede comprobarse contrastando los titulares de la prensa diaria con los "topics" que están siendo más mencionados durante un periodo de tiempo. Existen numerosos ejemplos en los que *Twit-*

ter se ha convertido en el canal principal de transmisión de la información sobre un determinado tema o hecho. Uno de los casos que alcanzó mayor repercusión fue la propagación de información relativa a las elecciones presidenciales de Irán en Junio de 2009 (*#IranElection*[40]).

En la Captura 1 puede visualizarse una página de perfil de un usuario arbitrario de *Twitter*. En ella se aprecian los distintos elementos que configuran toda página de perfil: nombre (1) e imagen del usuario (2), su última entrada (3), su historial de entradas (4), sus atributos (5), su número de amigos, seguidores, "tweets" etc. (6), las listas que ha creado (7), y por último una serie de enlaces a algunos de sus amigos mediante su imagen de perfil (8).

Captura 1: Perfil de usuario de Twitter

En lo que sigue, se utilizará la terminología propia del servicio de red social *Twitter* compuesta principalmente por los siguientes términos:

- **Entrada (“tweet”)**: mensaje de texto de 140 caracteres como máximo que es dado a conocer públicamente o notificado en exclusiva a los amigos.
- **Amigo (“following”)**: usuario de *Twitter* que notifica sus nuevas entradas a otro en tiempo real. En función de la configuración de privacidad, puede requerirse el consentimiento previo del primero para establecer la relación.
- **Seguidor (“follower”)**: usuario de *Twitter* que es notificado de las nuevas entradas de otro en tiempo real.
- **Re-entrada (“retweet”)**: consiste en reproducir íntegramente el mensaje de otro usuario, incluyendo su nombre y la palabra “retweet” o RT al inicio.
- **Respuesta (“reply”)**: respuesta al “tweet” de otro usuario de forma pública.
- **Mención (“mention”)**: el signo @ seguido por un nombre de usuario permite a los usuarios mencionar a otros en sus “tweets”.
- **Mensaje dirigido (“direct message”)**: forma privada de comunicación entre usuarios de *Twitter*.
- **Tema (“topic” o “hashtag”)**: palabras o frases con un prefijo #. Normalmente los temas con mayores menciones diarias **(“trending topics”)**, suelen estar relacionadas con noticias de actualidad o acontecimientos recientes.
- **Listas (“lists”)**: permiten clasificar a los amigos *Twitter* en grupos a voluntad del usuario, para poder seguirlos de forma más fácil.
- **Usuario popular (“twitterati”)**: usuario muy leído o influyente que posee un gran número de seguidores. En general suele tratarse de personajes públicos o celebridades, por ejemplo *Barack Obama*[41] o *Lady Gaga*[42].

El uso de *Twitter* es muy sencillo e intuitivo, y constituye sin duda uno de los factores de su rápida expansión. Presenta una interfaz de usuario a la que se puede acceder desde distintos tipos de dispositivos, navegadores y aplicaciones. En cuanto a su disponibilidad en distintos idiomas, *Twitter* fue creado en un principio en inglés exclusivamente para dar soporte más tarde también al japonés. Recientemente, y si-

guiendo una estrategia de captación de nuevos clientes en otras zonas del mundo, se han incluido los idiomas español, francés, italiano y alemán.

Twitter dispone de una API pública que se describe en el siguiente subapartado y permite acceder a su información de una forma práctica y poco costosa.

2.1.3. API de Twitter

La API[43] de *Twitter* es relativamente potente y extensa, además de estar abundantemente documentada. Sus principales inconvenientes son las limitaciones que impone *Twitter* sobre su uso y son descritas en el subapartado 5.1.1. La API está compuesta por tres módulos diferentes: dos que siguen el paradigma de programación REST (*REST API* y *Search API*) y otro más reciente basado en "streaming". Esta división se debe a motivos históricos, ya que todos los módulos han sido desarrollados de forma independiente y más tarde integrados, o aún en proceso, bajo una única API.

Los métodos de la *REST API* de *Twitter* permiten a los desarrolladores acceder al núcleo de los datos del servicio de red social. Esto significa que cualquier acción posible a través del portal web, puede realizarse también mediante la API. Adicionalmente la *Search API* ofrece a los desarrolladores métodos para interactuar con *Twitter Search* e información relativa a tendencias en *Twitter*. Por su parte, la API de "streaming" proporciona un elevado volumen de acceso a los "tweets" en tiempo prácticamente real, y de forma filtrada. Desde el punto de vista del desarrollador la división de *Twitter* en tres módulos es transparente, excepto por las limitaciones de uso de cada módulo y sus formatos de representación de la información.

Actualmente el proceso de integración de los módulos continúa y se están comenzando a aplicar medidas como la unificación de las limitaciones de uso de cada uno de ellos. Es un proceso dinámico y en constante avance, que puede seguirse a través del perfil de *Twitter twitterapi*[44].

2.2. Cloud Computing

En este subapartado se aborda el paradigma de "cloud computing" o computación en nube, y en particular se describe la plataforma *Google App Engine* (*GAE*) basada en dicho paradigma.

2.2.1. Paradigma

La computación en nube, del inglés "cloud computing", es un paradigma que permite ofrecer servicios de computación a través de Internet, de forma transparente para el cliente en términos de infraestructura. En este tipo de computación todo lo que puede ofrecer un sistema informático se hace como servicio desde la nube de Internet, de modo que los clientes y desarrolladores puedan acceder a ellos sin necesidad de invertir en potentes infraestructuras informáticas propias y abstrayéndose de la gestión de los recursos físicos empleados. La mayoría de los proveedores de "cloud computing" ofrecen servicios escalables dinámicamente que hacen uso de recursos virtuales, y son facturados siguiendo un modelo clásico de suministro de servicios: el usuario paga por la cantidad y calidad de los servicios disfrutados.

El paradigma "cloud computing" surge como una evolución lógica de los servicios de computación y almacenamiento de datos. De la misma manera que cien años atrás el consumo de energía eléctrica pasó a ser cubierto por las grandes compañías de abastecimiento de la red eléctrica, los proveedores de servicios "cloud computing" representan en cierta forma algo similar, pero relativo al proceso de aplicaciones y datos. En este contexto durante las últimas décadas se ha evolucionado desde los servidores en CPDs propios hasta el paradigma "cloud computing", pasando por la externalización o "hosting" de servidores. Durante dicha evolución, es necesario mencionar también como alternativa al "cloud computing" el paradigma "grid computing", que propugna la utilización de recursos heterogéneos no sujetos a un control central de forma coordinada. De hecho, ambos movimientos cuentan con igual número de segui-

dores y detractores, si bien ciertos autores[45] consideran que el "cloud computing" no es más que un tipo de "grid computing", dado que se trata de un paradigma que surgió con cierta posterioridad. La Figura 3 muestra cada una de las etapas mencionadas, junto con sus principales características.

Figura 3: Evolución de los servicios de computación y almacenamiento de datos

El término "cloud computing" es un concepto general que incorpora el software (*SaaS*), la plataforma (*PaaS*) o la infraestructura (*IaaS*) como servicio, tal como la *Web 2.0* y otros movimientos recientes promueven. El denominador común de todos ellos es la confianza en Internet para satisfacer las necesidades de los usuarios. En este sentido han aparecido también durante los últimos años algunos movimientos contrarios, basados en el hecho de que el paradigma "cloud computing" no permite a los usuarios disponer físicamente de sus datos, a excepción de copias voluntarias, y delega su control exclusivamente en manos del proveedor.

Dentro de un modelo de "cloud computing" se distinguen varias capas entre el cliente y la infraestructura, tal y como se observa en la Figura 4.

- **Cliente**: componentes hardware y/o sistema software que hacen uso de los servicios ofrecidos por la nube, o en caso estricto han sido diseñados en ex-

clusiva con ese fin y no tienen utilidad en otro contexto. Por ejemplo: clientes móviles (terminales móviles), clientes ligeros (sistemas operativos basados en "cloud computing") y clientes pesados (navegadores web).

- **Aplicación**: actúa como enlace entre el cliente y la plataforma, ofreciendo los servicios al usuario y gestionando los recursos empleados en la plataforma. En muchos casos se trata de aplicaciones que no requieren instalación ni ejecución en el lado del cliente, evitando las tareas relacionadas con la instalación, mantenimiento y soporte del software.
- **Plataforma**: ofrece una plataforma de computación y/o solución de pila como un servicio. Facilita el despliegue de aplicaciones sin el coste y la complejidad de comprar y gestionar el hardware subyacente y las capas de software.
- **Infraestructura**: recursos físicos empleados por la plataforma y con ubicación desconocida para el cliente. En muchos casos se encuentran agrupados en grandes centros de computación.

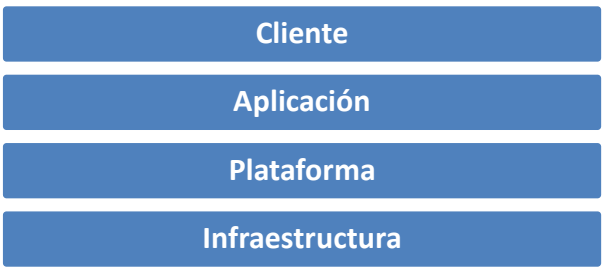


Figura 4: Capas del modelo de "cloud computing"

Como ejemplos de servicios de "cloud computing" públicos destacan: *Google App Engine* [5], *Amazon EC2*[21] y *Windows Azure*[22], que proveen aplicaciones comunes en línea accesibles desde distintos dispositivos, mientras el software y los datos se almacenan en los servidores. Todos ellos se basan en el mismo paradigma, pese a que cada uno posee sus particularidades y en algunos casos existen diferencias notables entre ellos. Por ejemplo los dos servicios más populares, *Google App Engine* y *Amazon EC2*, difieren en el modo en el que las aplicaciones desarrolladas son migradas

a la infraestructura física. *GAE* realiza esta migración de forma automática, en función de las necesidades dinámicas de las ejecuciones de la aplicación, mientras que con *Amazon EC2* es el mismo desarrollador el encargado del mapeo físico de sus aplicaciones. Esto último permite un mayor grado control, pero al mismo tiempo implica una mayor carga de trabajo.

2.2.2. Google App Engine (GAE)

Google App Engine (*GAE*)[5] es una plataforma concebida para desarrollar, alojar y ejecutar aplicaciones web sobre la infraestructura *Google*[46]. La plataforma hace uso del paradigma "cloud computing", mediante la virtualización de aplicaciones a través los numerosos servidores de los centros de datos de *Google*, diseminados por todo el mundo.

La infraestructura *Google* es totalmente transparente para el cliente de los servicios "cloud computing", quien se despreocupa de la gestión de los recursos utilizados, mientras que el usuario desarrollador por su parte es capaz de crear, mantener y actualizar sus aplicaciones. Se entiende como usuario desarrollador al programador de aplicaciones sobre la plataforma *GAE*, para que más tarde éstas sean ejecutadas por los clientes de los servicios. Al contrario que plataformas como *Amazon EC2,* que virtualizan a nivel de imágenes de máquinas virtuales, *GAE* ofrece su infraestructura para contener aplicaciones exclusivamente.

GAE ejecuta de forma eficiente aplicaciones escalables gracias a una entidad llamada *Balanceador de Carga*, que se encarga de asignar más o menos recursos (servidores) a cada una de ellas. El esquema del funcionamiento de *Google App Engine* es el siguiente: clientes acceden a las aplicaciones, que son implementadas por un número de servidores determinado por el *Balanceador de Carga*, en función de los recursos necesarios y/o disponibles. Por su parte, los servidores se sirven de la API que les es proporcionada por *Google* y al mismo tiempo también pueden hacer uso de otros ser-

vicios de la infraestructura *Google* como la *BigTable* o las cuentas de usuario (*Google Accounts*). La Figura 5 representa gráficamente la arquitectura de la plataforma para diferentes tipos de cliente o peticiones entrantes, como por ejemplo navegadores o dispositivos móviles:

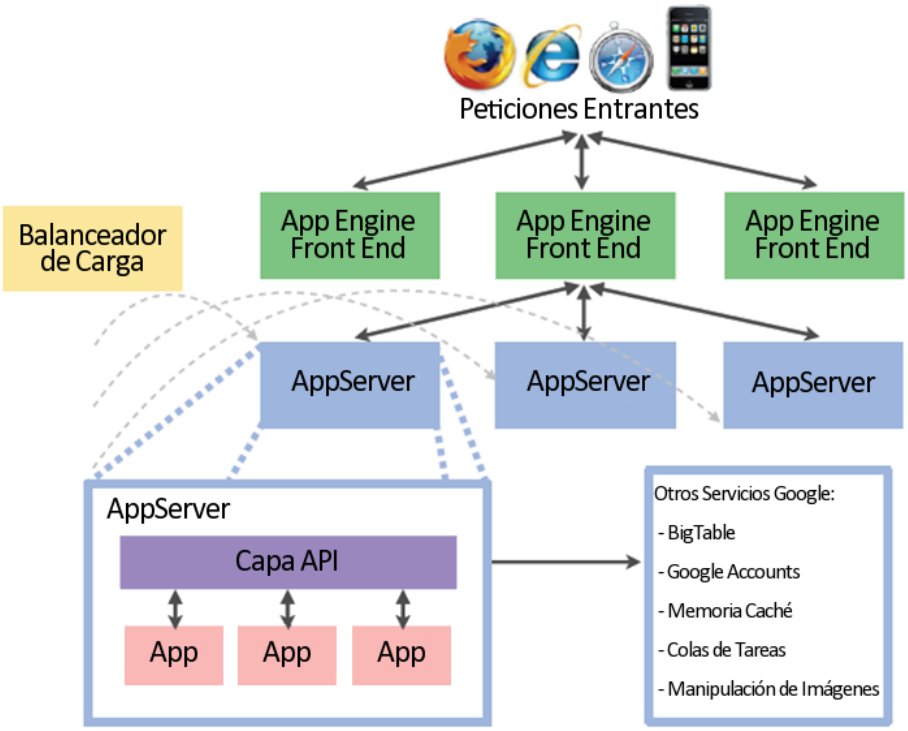


Figura 5: Arquitectura de Google App Engine (GAE)

Desde el punto de vista de los clientes de servicios, y no de los usuarios desarrolladores, el uso de *GAE* no requiere la instalación de ningún programa especifico (se trata de aplicaciones web accesibles desde un navegador cualquiera), y es totalmente gratuito y transparente. Por su parte, el usuario desarrollador es el encargado de habilitar una cantidad determinada de recursos disponibles para sus aplicaciones. Existe una cuota de recursos gratuita, pero si se desea permitir el consumo de recursos adicionales, es necesario activar una cuenta de facturación y en ese caso el usuario desarrollador pagará por todos aquellos recursos utilizados por los clientes de servicios, que excedan los límites de las cuentas gratuitas. Los recursos que se encuentran

disponibles van desde capacidad de almacenamiento extra, mayor ancho de banda o tiempo de CPU superior. Los límites por recurso de las cuentas gratuitas de *GAE* pueden consultarse en el subapartado 5.1.2. Por otra parte, a día de hoy, por cada cuenta de usuario *Google* no es posible alojar más de 10 aplicaciones diferentes.

GAE fue lanzado en Abril de 2008 en versión beta con soporte en exclusiva para *Python*[47], y más tarde incorporó también el lenguaje de programación *Java*[48]. Sin embrago, *Google* impone una serie de restricciones en el uso de dicho lenguaje: los hilos, las conexiones de red directas y el código nativo, no están habilitados. Actualmente *GAE* se encuentra en constante desarrollo, con un número creciente de funcionalidades ofrecidas. Además, *Google* planea dar soporte a más lenguajes en el futuro ya que concibe la plataforma como independiente del lenguaje.

2.3. Teoría de Grafos

En el ámbito de las ciencias matemáticas, la teoría de grafos estudia las propiedades de los grafos y las conclusiones que pueden obtenerse del análisis de su estructura y de su composición. En los siguientes subapartados, se procede a exponer la importancia del análisis de los grafos sociales y a describir las métricas más utilizadas en dicho campo.

2.3.1. Análisis de Grafos

El análisis de grafos sociales, y su relación con la teoría de grafos, se ha convertido durante los últimos años en una herramienta muy valiosa para la sociología moderna y en un tema de gran interés para la investigación y el estudio. También ha ganado peso en otros muchos campos como la antropología, la biología, el periodismo, la economía o la psicología. En conclusión, el análisis de grafos sociales ha pasado de ser una prometedora metáfora a una ciencia analítica, con sus propios teoremas, métodos, herramientas software e investigadores.

Históricamente han sido muchos los investigadores que se han interesado por el análisis de grafos sociales, con el objeto de extraer conclusiones acerca del comportamiento de sus miembros. Sin embargo, desde el inicio de dichos estudios se han planteado ciertos problemas que han limitado la potencia de sus resultados. El primero de ellos fue la dificultad para obtener grafos adecuados para el análisis. La cantidad de datos necesarios para construir un grafo social lo suficientemente extenso y no sesgado del que extraer conclusiones fiables, es muy grande, y hasta hace no demasiado tiempo no se disponía de herramientas apropiadas para ello. Esto explica que en muchos estudios, el concepto de red social se haya empleado exclusivamente para denotar patrones entre vínculos relativamente cercanos y agrupaciones ordinarias, tales como lazos familiares, la pertenencia a una misma categoría social, etc.

Con la aparición de los servicios de redes sociales en Internet esta limitación desapareció, el concepto de red social se desarrolló profundamente, y los investigadores consiguieron una herramienta realmente útil para la obtención de grafos sociales complejos, plagados de entidades y de relaciones entre ellas de todo tipo. Sin embargo, en ese momento se planteó el siguiente gran problema y que aún sigue vigente en la actualidad: la dificultad del análisis de la información obtenida por su alto grado de complejidad. Este factor ha sido determinante en la no excesiva propagación del uso de estas nuevas herramientas con fines comerciales, ya que solo están al alcance de grandes empresas con relativamente elevados presupuestos.

Actualmente el concepto de red social se utiliza para definir complejos conjuntos de relaciones, desde interpersonales a suprapersonales, entre miembros de todos los sistemas y escalas sociales. El tamaño de una red social ayuda a determinar la utilidad del análisis para sus miembros. Las redes cerradas, densas y pequeñas, pueden ser menos útiles que las redes abiertas, con un gran número de relaciones con entidades fuera de la red principal. Este tipo de redes ofrecen más ideas y oportunidades para sus miembros que las primeras, excesivamente redundantes. En otras palabras, un

grupo de individuos aislado e independiente que comparte los mismos conocimientos, dispone de un acceso a la información global más limitado que otro grupo más abierto. Por otra parte, la influencia y repercusión de los actos es mayor en la red abierta, a pesar de que en la red cerrada puede llegar a ser más efectiva.

El fenómeno de "small world" es la hipótesis de que la cadena de relaciones sociales necesarias para conectar a una persona con otra arbitraria en cualquier parte del mundo es generalmente corta. El concepto dio lugar a la famosa frase de seis grados de separación, que surgió después de que el psicólogo *Stanley Milgram* realizara el experimento del "small world"[49] en 1967. Durante el experimento se tomó una muestra de individuos y se pidió hacer llegar un mensaje a una persona objetivo en particular, a lo largo de una cadena de conocidos. La longitud media de las cadenas de éxito resultó ser de unos cinco o seis pasos intermedios de separación, a pesar de que la mayoría de las cadenas del estudio, alrededor del 80%, no alcanzaron el objetivo.

En la actualidad los investigadores siguen estudiando este fenómeno, pero teniendo en cuenta nuevos canales comunicativos como Internet, y no solo los sistemas postales o el teléfono disponibles en la época de *Milgram*. Por ejemplo, un reciente experimento[50] desarrollado en la *Universidad de Columbia* similar al "small world", pero empleando el correo electrónico, encontró que de cinco a siete grados de separación son suficientes para conectar a dos personas a través de e-mail en todo el mundo.

2.3.2. Métricas de Análisis

Dentro de la teoría de redes y del análisis de grafos, existen ciertos conjuntos de medidas denominadas de caracterización de un vértice que determinan su importancia relativa en el grafo, como por ejemplo, la importancia de una persona involucrada en una red social. Entre todas ellas destaca la centralidad, como atributo estructural de los nodos en una red.

A continuación son descritas algunas de las métricas de centralidad más empleadas, cuyas definiciones han sido extraídas del libro "*Small Worlds: The Dynamics of Networks between Order and Randomness*" de *Duncan J. Watts*[51]:

- **Radio ($\mathbf{rad}$)**: equivale a la mínima distancia entre cualquier par de nodos.
- **Diámetro ($\mathbf{D}$)**: al contrario que el radio, representa la máxima distancia entre cualquier par de nodos.
- **Grado nodal o Grado de un vértice ($\mathbf{v}$, $\mathbf{K_v}$)**: se trata de la primera y más simple definición de centralidad. Se define como el número de aristas que inciden en un vértice. El grado se interpreta a menudo como el número de conexiones que posee una persona en una red social.
- **Grado medio de un grafo ($\mathbf{K}$)**: se considera el valor del grado medio de todos los vértices que componen el grafo.
- **Distancia o camino más corto entre dos nodos ($\mathbf{d(i,j)}$)**: representa el número mínimo de vértices que han atravesarse para desplazarse desde un vértice i hasta otro vértice j.
- ***Caracteristic path length* ($\mathbf{L}$)**: se corresponde con la mediana de las medias de las distancias conectan cada vértice de un grafo con todos los demás.
- ***Local clustering coefficient* o coeficiente de agrupamiento local ($\mathbf{C_i}$)**: se trata de una métrica relativa a un vértice y viene dada por la proporción de enlaces existentes entre sus vértices vecinos, dividido por el número total de enlaces que podrían existir entre ellos.
- ***Network average clustering coefficient* o coeficiente de agrupamiento medio ($\mathbf{C}$)**: se refiere a todo el grafo y equivale a la media los *local clustering coefficients* de todos los vértices.

Como puede suponerse existen infinidad de métricas diferentes además de las recién descritas, sin embargo únicamente se ha querido hacer hincapié en aquellas relacionadas con la fase de análisis de los grafos sociales extraídos durante el proyecto.

3. Arquitectura del Sistema

En el siguiente apartado de la memoria se realiza una descripción del esquema general de la arquitectura del sistema, para más tarde detallar los distintos entornos de ejecución con sus aplicaciones correspondientes.

La arquitectura del extractor de información de *Twitter* está compuesta tanto por componentes desplegados de forma local, como por componentes desplegados en el entorno "cloud". El uso de este tipo de entorno se justifica por la capacidad que ofrece *Google App Engine* de ejecutar una misma aplicación desde un gran número de máquinas físicas diferentes, y de esta forma sortear las restricciones sobre el uso de la API de *Twitter* que son descritas en el subapartado 5.1.1. Por otra parte, la arquitectura en el entorno local permite llevar un control y un acceso más directo sobre los datos extraídos, para ser analizados desde una máquina concreta en la que hayan sido descargados y almacenados previamente.

3.1. Esquema General

Atendiendo a una visión general de la arquitectura del sistema, pueden distinguirse tres entornos de ejecución diferentes:

- **Entorno local**: agrupa las aplicaciones ejecutadas desde una máquina local determinada, escogida para desencadenar todo el proceso de extracción y finalmente recopilar los datos para el análisis. Forman parte del entorno local el generador del conjunto raíz, el lanzador de peticiones, el recuperador de datos, el seguidor de usuarios y la base de datos local.
- **Entorno "cloud" (*Google App Engine*)**: reúne el conjunto de extractores de amigos y de seguidores ejecutados desde distintas máquinas físicas asignadas por *Google*. A efectos prácticos se trata de dos tipos de aplicaciones li-

geramente diferentes, replicadas cinco veces cada una de ellas con el fin de distribuir el proceso de extracción.

- ***Twitter***: representa los servicios ofrecidos por el servicio de red *Twitter* a través de su *REST API*. Se trata de la fuente de información de la que se alimenta el sistema, para poder analizarla posteriormente en el entorno local.

La arquitectura del sistema ha sido adaptada para permitir la ejecución de determinados componentes (extractores de amigos y de seguidores) en el entorno "cloud", donde cada aplicación es ejecutada en una máquina diferente según determine el *Balanceador de Carga* de *Google App Engine*. Dichas máquinas pertenecen a alguno de los centros de computación de *Google* diseminados por todo el mundo, y esta asignación de recursos físicos por aplicación es totalmente transparente para el cliente de los servicios "cloud".

La transferencia de información entre el entorno local y *GAE* se lleva a cabo mediante el protocolo HTTP. Los componentes desplegados en *GAE* funcionan como aplicaciones web que recogen las peticiones del entorno local, ya sean de planificación de tareas o de descarga de datos. En consecuencia el tráfico de información entre los dos entornos es bidireccional, dependiendo de la fase en la que se encuentre el proceso. De la misma manera, *GAE* interacciona con el servicio de red social de *Twitter* mediante peticiones HTTP, pero en este caso el proceso se gestiona de forma interna por la librería *Twitter4J*[52]. De cualquier modo, toda transferencia de información realizada en el sistema es registrada por los logs de ejecución de las aplicaciones del entorno local, o por la *Consola de Administración* de *GAE* para los componentes desplegados en la infraestructura de "cloud computing".

La Figura 6 muestra una representación gráfica del esquema general de la arquitectura del sistema, con los tres entornos de ejecución diferentes. Las flechas bidireccionales entre ellos representan el tráfico de información de unos entornos a otros.

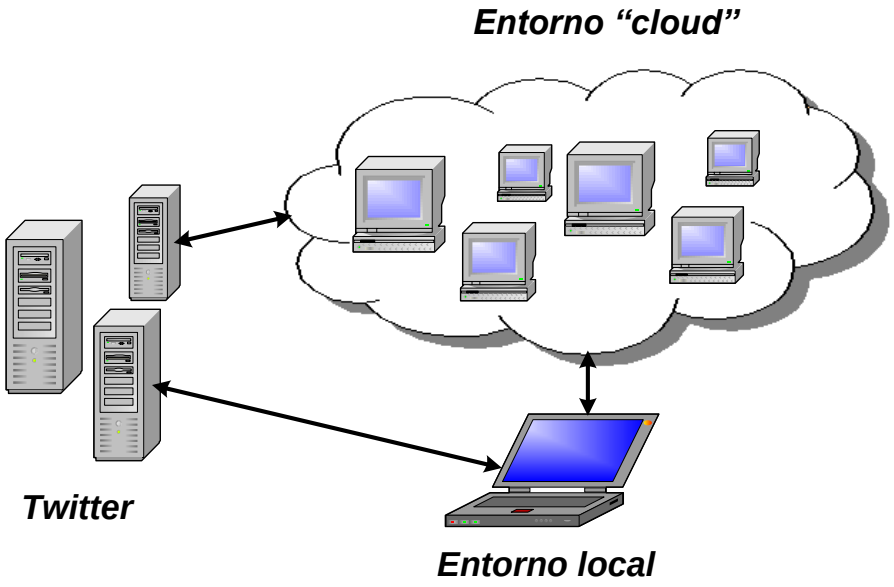


Figura 6: Esquema general de la arquitectura del sistema

Como puede observarse en la figura, el entorno "cloud" o *GAE* actúa como mera capa intermedia entre *Twitter* y el entorno local para sortear las restricciones sobre el uso de la API de *Twitter* durante determinadas fases del proceso.

3.2. Entorno Local

El entorno local incluye el mayor número de aplicaciones del sistema. Los siguientes subapartados describen en detalle cada una de sus aplicaciones, incluyendo representaciones gráficas en forma de diagramas UML.

3.2.1. Generador del Conjunto Raíz

El generador del conjunto raíz es el encargado de seleccionar el conjunto de usuarios de *Twitter* sobre el que realizar las extracciones periódicas de información. A pesar de requerir los servicios del entorno *Twitter*, pertenece al entorno local porque se emplea durante una fase del proceso en la que no se demandan excesivas peticiones de información, y por lo tanto no es crítica en cuanto a las restricciones de la API de *Twitter*. En consecuencia, y como excepción junto con la aplicación del seguidor de

usuarios, las peticiones de información se realizan de forma directa sin que sea necesario sortear las restricciones del servicio.

La siguiente figura muestra gráficamente la arquitectura del generador del conjunto raíz dentro del entorno local, y su conexión con el entorno *Twitter*:

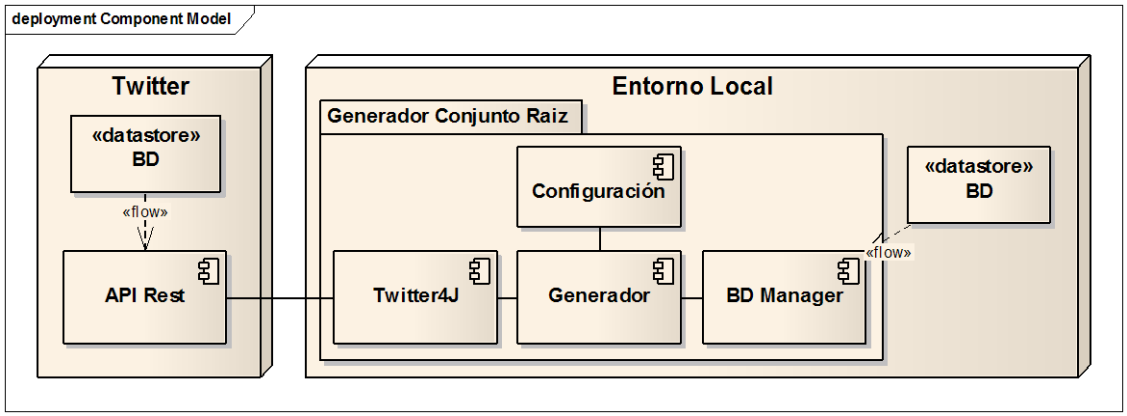


Figura 7: Arquitectura del generador del conjunto raíz

Los módulos que componen el generador del conjunto raíz son los siguientes:

- **Generador**: orquesta la generación del conjunto raíz a partir de un usuario semilla siguiendo un algoritmo de búsqueda en anchura. Para realizar las peticiones de información a *Twitter* sobre los usuarios a incluir en el conjunto, emplea la librería *Twitter4J*.
- ***Twitter4J*:** sirve de nexo de unión entre el generador y la *REST API* de *Twitter*. Esta librería facilita el proceso de consulta y de gestión de la información del servicio de red social, de forma transparente para el generador.
- **Configuración:** incluye parámetros de configuración que determinan la forma en que se genera el conjunto raíz (usuario semilla, tamaño máximo del conjunto, grado de propagación de las relaciones, etc.). Asimismo, contiene los parámetros de conexión con la base de datos local.
- **BD Manager:** gestiona la conexión con la base de datos local siguiendo unos parámetros de configuración.

Dado que tanto el generador del conjunto raíz, como el lanzador de peticiones pertenecen al entorno local, no es necesario contemplar ningún proceso de transferencia de información entre ellos como podría suponer la descarga de datos desde el entorno "cloud".

3.2.2. Lanzador de Peticiones

El lanzador de peticiones se ocupa de realizar las distintas peticiones HTTP desde el entorno local, para desencadenar las extracciones de información de *Twitter* mediante el entorno "cloud". El emplazamiento del lanzador de peticiones en el entorno local permite gestionar las peticiones de forma directa, en función de la información requerida por el conjunto raíz almacenado en la base de datos local.

La siguiente figura ilustra la arquitectura del lanzador de peticiones dentro del entorno local, y su conexión con el entorno "cloud":

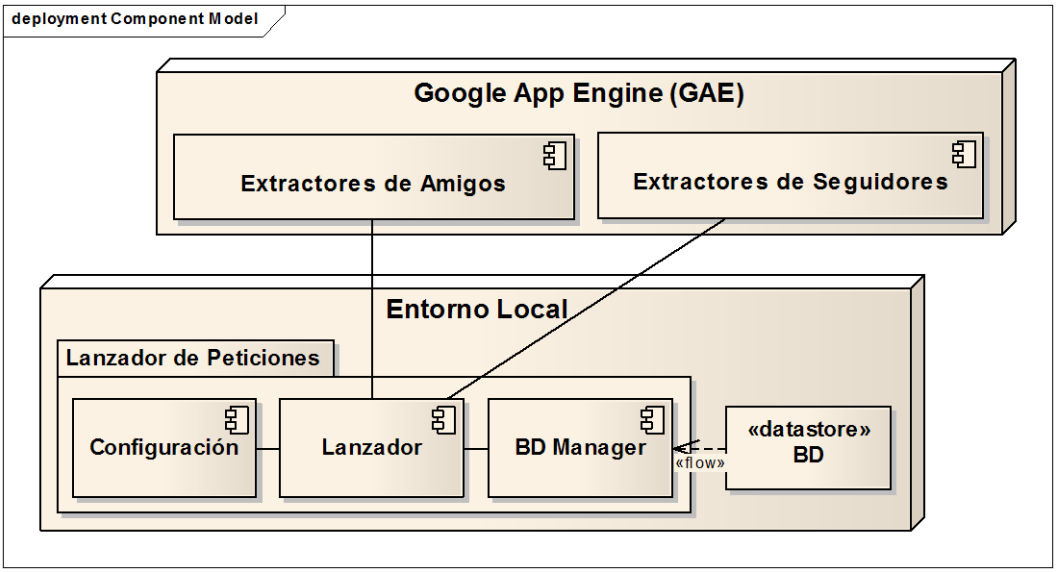


Figura 8: Arquitectura del lanzador de peticiones

Los componentes que conforman el lanzador de peticiones dentro del entorno local son los siguientes:

- **Lanzador**: realiza las distintas peticiones HTTP a los servidores de extracción siguiendo unos parámetros de configuración y en función de la composición del conjunto raíz, almacenado en la base de datos local.
- **Configuración:** incluye parámetros de configuración que determinan por su URL los servidores de extracción en el entorno "cloud", y los subconjuntos del conjunto raíz asignados a cada uno de ellos. Igualmente, contiene los parámetros de conexión con la base de datos local.
- **BD Manager:** gestiona la conexión con la base de datos local siguiendo unos parámetros de configuración.

El diseño del lanzador de peticiones permite su ejecución simultánea desde más de una máquina en el entorno local, con objeto de generar un mayor número de peticiones durante un periodo de tiempo. Sin embargo, para el tamaño del conjunto raíz generado y el número de extractores de información en el entorno "cloud" desplegados, un único lanzador de peticiones se ha considerado suficiente. En el caso de que se planteara la posibilidad de ejecutar en paralelo dos o más lanzadores de peticiones, los únicos requisitos serían dos: los lanzadores deberían poder acceder al mismo repositorio de información en el que se encontrará el conjunto raíz (base de datos, fichero, etc.) y las peticiones tendrían que ser divididas en conjuntos disjuntos.

3.2.3. Recuperador de Datos

El recuperador de datos es el encargado de realizar las distintas peticiones HTTP que demandan y recopilan la información previamente extraída de *Twitter*, por las aplicaciones del entorno "cloud".

La Figura 9 muestra gráficamente la arquitectura del recuperador de datos dentro del entorno local, y su conexión con el entorno "cloud":

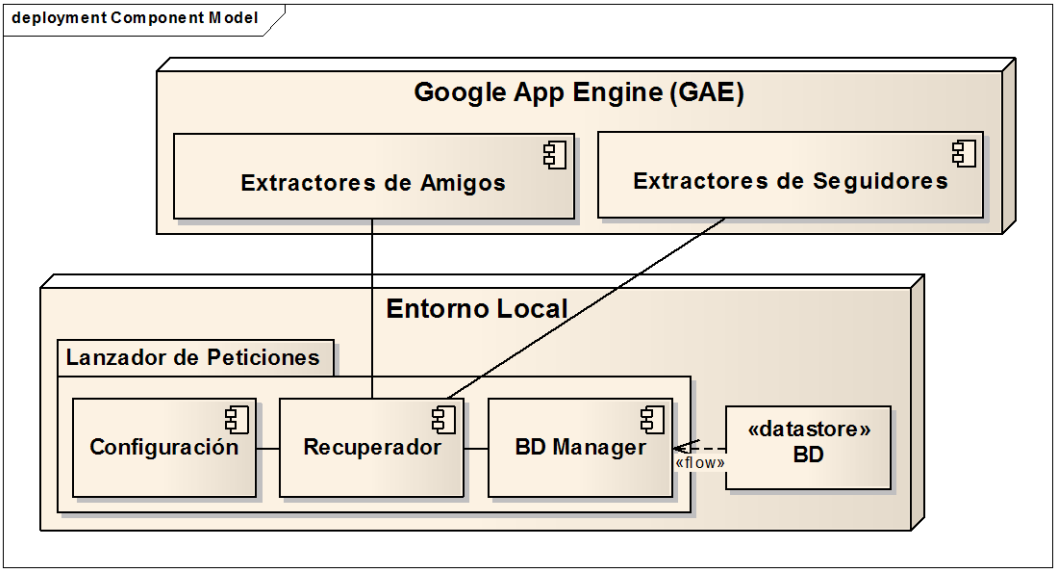


Figura 9: Arquitectura del recuperador de datos

Los módulos que forman parte del recuperador de datos son los siguientes:

- **Recuperador**: realiza las peticiones HTTP a los distintos servidores de extracción siguiendo unos parámetros de configuración, y en función de la composición del conjunto raíz almacenado en la base de datos local.
- **Configuración:** incluye las direcciones URL de los servidores *GAE* de dónde recuperar los datos, y los subconjuntos del conjunto raíz asignados a cada uno de ellos. Asimismo, contiene los parámetros de conexión con la base de datos local.
- **BD Manager:** gestiona la conexión con la base de datos local siguiendo unos parámetros de configuración.

Al igual que ocurre con el lanzador de peticiones, la recuperación de los datos podría paralelizarse mediante la ejecución simultánea de varios recuperadores en distintas máquinas. En este caso la complejidad sería ligeramente superior, dado que implicaría un acceso concurrente con escritura sobre el repositorio de información en el que se fueran almacenando los datos recuperados. Otra alternativa válida para la para-

lelización, sería separar los datos recuperados por cada aplicación en conjuntos disjuntos y fusionarlos al final del proceso.

3.2.4. Seguidor de Usuarios

El seguidor de usuarios se encarga de extraer periódicamente información acerca de un determinado y reducido conjunto de usuarios de *Twitter*. En consecuencia el número de peticiones HTTP necesarias es pequeño, por lo que estás pueden llevarse a cabo de forma directa, sin tener que recurrir al uso de *GAE* para sortear las restricciones del servicio de red social *Twitter*.

La siguiente figura representa gráficamente la arquitectura del seguidor de usuarios dentro del entorno local, y su conexión con el entorno *Twitter*:

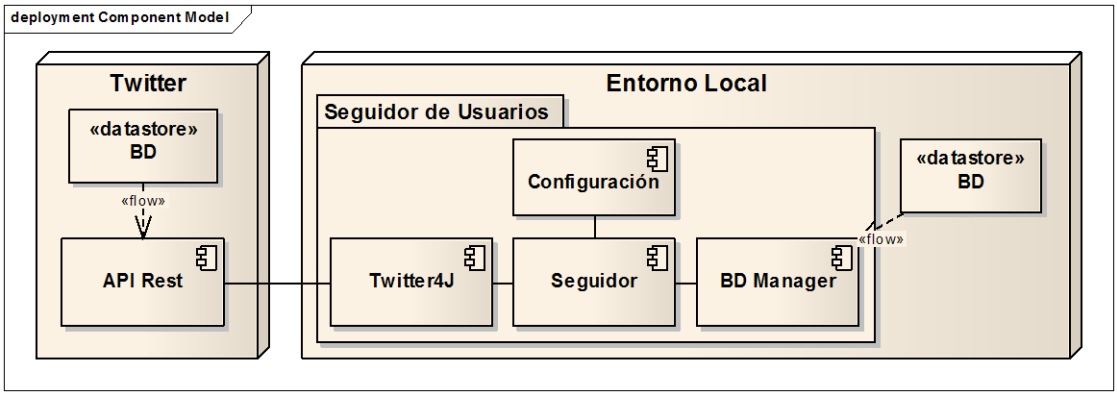


Figura 10: Arquitectura del seguidor de usuarios

Los módulos que conforman el seguidor de usuarios son los siguientes:

- **Seguidor**: realiza las peticiones de información periódicas sobre una serie de usuarios determinados. Para realizar las demandas de información a *Twitter* emplea la librería *Twitter4J*.
- ***Twitter4J*:** sirve de nexo de unión entre el seguidor de usuarios y la *REST API* de *Twitter*. Se ocupa del proceso de consulta y de gestión de la información de *Twitter*, de forma transparente para el seguidor.

- **Configuración:** incluye parámetros de configuración que determinan los usuarios del seguimiento y la información requerida de cada uno de ellos. Asimismo, contiene los parámetros de conexión con la base de datos local.
- **BD Manager:** gestiona la conexión con la base de datos local siguiendo unos parámetros de configuración.

De la misma forma que ocurre con el generador del conjunto raíz, el seguidor de usuarios no requiere ningún proceso de descarga de datos desde la infraestructura "cloud computing", por estar desplegado en el entorno local.

3.3. Entorno "Cloud"

El entorno "cloud" o *GAE* reúne el conjunto de extractores de amigos y de seguidores, que son ejecutados desde distintas máquinas físicas asignadas por *Google* mediante su *Balanceador de Carga*. Como ya se ha mencionado, este mecanismo le permite realizar peticiones de información a *Twitter* desde distintas direcciones IP, y así sortear sus restricciones del servicio. Por otra parte, la replicación de los extractores en el entorno "cloud" ha sido necesaria dadas las limitaciones de las cuentas gratuitas de *GAE,* sobre el tiempo de CPU y el espacio de almacenamiento por aplicación.

3.3.1. Extractor de Amigos/Seguidores

Los extractores de amigos y de seguidores realizan peticiones de información al servicio de red social *Twitter*, que se recogen de forma temporal en el almacén de datos de *GAE*. Asimismo, cuando dicha información es solicitada por el entorno local vía peticiones HTTP, estos permiten que sea descargada en varios formatos. Por último, ofrecen adicionalmente la posibilidad de borrar el contenido del almacén de datos.

La Figura 11 ilustra la arquitectura de un extractor de amigos o de seguidores tipo dentro *GAE*, y su conexión con *Twitter*:

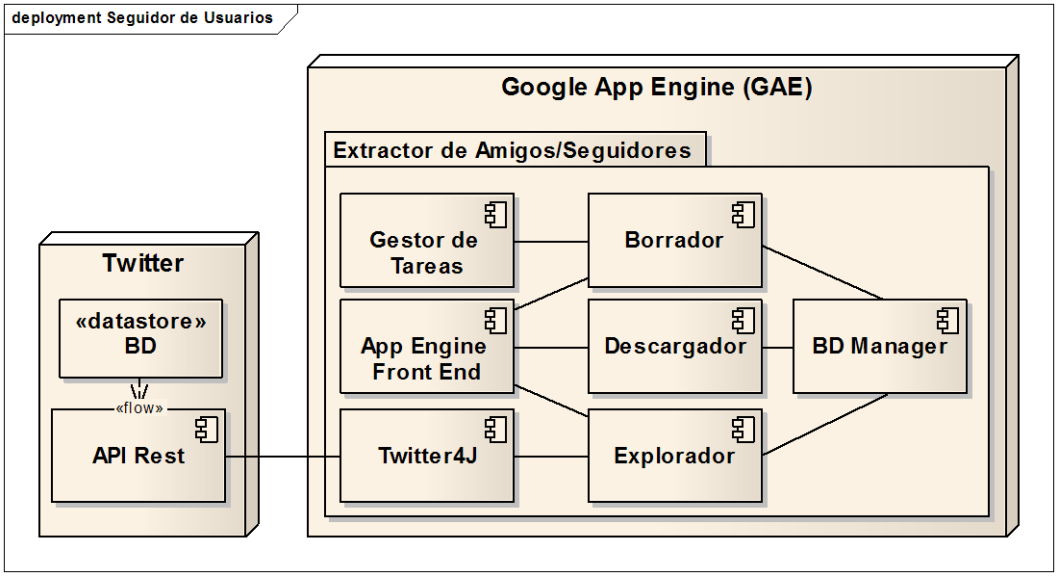


Figura 11: Arquitectura del extractor de amigos/seguidores

Los módulos que conforman un extractor de amigos o de seguidores tipo son los siguientes:

- ***App Engine Front End*:** se ocupa de las interacciones con el entorno local. Dado que *GAE* funciona como una plataforma de aplicaciones web, su ejecución solo puede desencadenarse mediante peticiones HTTP, que son procesadas por clases que implementan la interfaz "servlet".
- **Gestor de tareas**: genera una serie de tareas que son añadidas a sus respectivas colas, a partir de los distintos tipos de peticiones procesadas por el *App Engine Front End*.
- **Explorador:** ejecuta las tareas de petición y almacenamiento de información al servicio de red social *Twitter* acerca de los amigos o seguidores de un usuario determinado. Para realizar las demandas de información a *Twitter* emplea la librería *Twitter4J*.
- ***Twitter4J*:** sirve de nexo de unión entre el explorador y la *REST API* de *Twitter*. Se ocupa del proceso de consulta y de gestión de la información de *Twitter*, de forma transparente para el explorador.

- **Descargador:** ejecuta las tareas de descarga de los datos recogidos en el almacén de datos de *GAE*, facilitándolos en varios formatos.
- **Borrador:** ejecuta las tareas de borrado de la información contenida en el almacén de datos *GAE*.
- **BD Manager:** gestiona la conexión con el almacén de datos de *GAE* siguiendo unos parámetros de configuración.

Adicionalmente, los extractores de amigos y seguidores incluyen de forma implícita el *Balanceador de Carga* de *GAE* que permite que puedan ser ejecutados desde distintas máquinas físicas asignadas por *Google*.

4. Desarrollo, Monitorización y Análisis del Sistema

En el cuarto apartado de la memoria se describen todas las plataformas, lenguajes, y herramientas empleadas en las distintas fases de desarrollo, monitorización y análisis del sistema. A continuación se describen sus características más importantes, prestando especial atención a aquellos elementos menos comunes.

4.1. Plataformas

La naturaleza mixta de la arquitectura del sistema descrita en el apartado anterior, así como ciertas herramientas exclusivas, han determinado el uso de varias plataformas informáticas durante las distintas fases del proyecto. En concreto, las plataformas son las siguientes:

- ***Microsoft Windows Vista***: utilizada tanto para el desarrollo del sistema mediante el entorno de desarrollo integrado *Eclipse* y la monitorización de sus ejecuciones, como para la redacción de la memoria.
- ***GNU/Linux***: empleada exclusivamente para el análisis de la información extraída y su posterior representación gráfica, mediante las herramientas *Graph-Tool* de análisis de grafos y *Pylab* de generación de gráficas.
- ***Google App Engine* (*GAE*)**: encargada de alojar y servir de entorno de ejecución para las aplicaciones "cloud computing" del sistema.

De las tres plataformas recién mencionadas, únicamente el uso de dos de ellas ha sido imprescindible en el desarrollo, monitorización y análisis del sistema. Es el caso de *GAE*, entorno de ejecución exclusivo para las aplicaciones "cloud computing", y *GNU/Linux*, determinado por la herramienta de análisis de grafos *Graph-Tool*. *Microsoft Windows Vista*, por su parte, ha sido escogido por motivos prácticos, aunque no determinantes, para la realización del resto de etapas.

4.2. Lenguajes de Programación y Consulta

Uno de los indicadores de la complejidad del sistema desarrollado viene representado por el elevado número de lenguajes de programación y consulta presentes en alguna de las fases del proyecto. En particular, los lenguajes de programación utilizados en la codificación del sistema han sido *Java* y *Python*, mientras que para las tareas de consulta se han empleado *SQL* y *GQL*.

4.2.1. Java

Java[48] es un lenguaje de programación orientado a objetos ampliamente extendido, desarrollado por *Sun Microsystems* a principios de los años 90. En la actualidad, *Java* ha pasado a ser propiedad de *Oracle Corporation*. El lenguaje en sí mismo toma mucha de su sintaxis de *C* y *C++*, pero tiene un modelo de objetos más simple y elimina la gestión a bajo nivel de los elementos, que suele inducir a muchos errores, como la manipulación directa de punteros o memoria.

GAE permite lanzar aplicaciones en dos entornos de ejecución diferentes: el entorno *Java* y el entorno *Python*. Cada uno de ellos proporciona una serie de protocolos estándar y tecnologías propias para el desarrollo de aplicaciones web, como por ejemplo las *Java Server Pages* (*JSP*) en *Java*. A pesar de haber aparecido con anterioridad el entorno *Python*, la popularidad del lenguaje *Java* ha provocado que sus versiones del "binding" o adaptador para *GAE* hayan alcanzado un alto grado de convergencia, y ofrezcan en estos momentos prácticamente las mismas funcionalidades.

Para el desarrollo de las aplicaciones de extracción en el entorno "cloud" se ha optado por utilizar el entorno de ejecución *Java*, puesto que ya se disponía de unos conocimientos previos muy avanzados de dicho lenguaje.

4.2.2. Python

Python[47] es un lenguaje de programación interpretado creado por *Guido van Rossum* en el año 1991. En la actualidad se desarrolla como un proyecto de código abierto, administrado por la *Python Software Foundation*. *Python* se utiliza como lenguaje de programación interpretado, lo que ahorra un tiempo considerable en el desarrollo del programa, pues no es necesario compilar ni enlazar.

Con objeto de poder hacer uso del módulo de *Python* para el análisis de grafos, *Graph-Tool*, se ha empleado dicho lenguaje de programación. En concreto la versión estable utilizada ha sido *Python 2.6.4*.

4.2.3. SQL

El lenguaje de consulta estructurado (*SQL*) es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones sobre éstas. Fue comercializado por primera vez en 1981 por *IBM* y desde entonces ha sido considerado como un estándar para las bases de datos relacionales. Una de sus características es el manejo del álgebra y el cálculo relacional para efectuar consultas, con el fin de recuperar la información de una forma sencilla, así como de realizar modificaciones sobre ella.

Desde el punto de vista del proyecto, la gestión del almacenamiento de los datos en el entorno local se ha llevado a cabo mediante instrucciones *SQL* sobre un sistema *MySQL*, descrito en profundidad en el subapartado 4.3.4.

4.2.4. GQL

GQL[53] es el lenguaje utilizado para recuperar entidades del almacén de datos escalable de *Google App Engine*. Aunque las funciones de *GQL* son distintas que las de un lenguaje de consulta de un sistema relacional tradicional, su sintaxis es similar a la

de *SQL*. Una de las principales diferencias entre ambos lenguajes es que el almacenamiento de datos en *GAE* es transaccional, se realiza de una forma no relacional en el sentido de *SQL*.

En lo referente a las limitaciones de *GQL*, el lenguaje no permite realizar consultas sobre más de una tabla de forma conjunta. Además, *GQL* no soporta sentencias JOIN, dado que éstas se muestran ineficientes cuando las consultas han de acceder a más de una máquina para recuperar los datos. En su lugar, relaciones uno-a-uno y muchos-a-muchos pueden implementarse utilizando propiedades especiales de referencia. Este particionamiento de la información permite aislar posibles errores en alguno de los sistemas de almacenamiento, evitando provocar un error general del sistema. Como inconveniente, se requiere un diseño específico del modelo de datos por parte de los desarrolladores.

Por otra parte, *Google App Engine* limita el máximo número de filas devueltas por consulta a 1.000. Esta restricción no afecta a las aplicaciones web diseñadas para ser utilizadas por los clientes, ya que éstas pueden emplear listas pre-calculadas, y rara vez requerirán mostrar más de 1.000 registros en una sola página. En su lugar, existen mecanismos de paginación y almacenamiento en caché. En cualquier caso, si una aplicación necesita utilizar más de 1.000 registros por operación, puede utilizar su propio software desde el cliente y encadenar consultas secuencialmente.

4.3. Herramientas de Desarrollo

Las herramientas empleadas para el desarrollo del sistema han sido numerosas. Todas ellas han simplificado las tareas de desarrollo como es de suponer, si bien en algunas ocasiones de forma especialmente notable. Es el caso del “plugin” de *GAE*, que ha automatizado el despliegue de las aplicaciones en el entorno “cloud”, y del “binding” para *Java Twitter4J*, que ha simplificado enormemente el proceso de extracción de información del servicio de red social *Twitter*.

4.3.1. Eclipse

Eclipse[54] es un entorno de desarrollo integrado (IDE de sus siglas en inglés) de código abierto multiplataforma para desarrolladores. Fue concebido originalmente por *IBM* como el sucesor de su familia de herramientas para *VisualAge*. Actualmente es propiedad de la *Eclipse Foundation*, una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto y un conjunto de productos complementarios, capacidades y servicios.

Las principales características de *Eclipse* son: dispone de un editor de texto con resaltado de sintaxis, la compilación es en tiempo real, posee pruebas unitarias con *JUnit*, control de versiones con *CVS*, integración con *Ant*, asistentes para creación de proyectos, clases, tests, etc., y refactorización. Asimismo, a través de infinidad "plugins" o complementos libremente disponibles es posible añadir nuevas funcionalidades como el control de versiones con *Subversion*.

El desarrollo del código del sistema se ha realizado mediante la herramienta *Eclipse*, por el gran número de ventajas anteriormente descritas. Igualmente, para la codificación de los módulos pertenecientes al entorno “cloud” se ha hecho uso del “plugin” para *GAE*.

4.3.2. GAE Plugin para Eclipse

El “plugin” de *GAE* para *Eclipse* permite desarrollar aplicaciones de *App Engine*, de forma ágil y eficaz. Incorpora unos botones especiales en la barra de herramientas, para facilitar la creación, compilación y despliegue de las aplicaciones.

Captura 2: Google App Engine Plugin para Eclipse

El complemento está desarrollado por *Google* y es objeto de frecuentes mejoras y actualizaciones. La versión utilizada en la codificación del proyecto ha sido la *Google Plugin for Eclipse 3.4*.

4.3.3. Twitter for Java (Twitter4J)

Uno de los principales problemas planteados a la hora de desarrollar el sistema de extracción de información ha sido la complejidad del proceso de interacción con el servicio de red social *Twitter*. La API de *Twitter* atiende peticiones HTTP, y devuelve las respuestas en formato XML que ha de ser tratado e interpretado. Afortunadamente, existen numerosos "bindings" ya desarrollados que se ocupan de simplificar este proceso. En particular, los más populares en el lenguaje *Java* son los siguientes:

- ***Twitter4J***[52].
- ***Java-Twitter***[55].
- ***JTwitter***[56].
- ***TweetStream4J***[57].

De todas ellas, *Twitter4J* es la más completa en cuanto a funcionalidades, estabilidad y frecuencia de actualizaciones. Soporta la mayoría de los métodos de los módulos de la API de *Twitter* (*REST API*, *Search API* y "streaming"), funciona como una biblioteca *Java* (*.jar*) y no posee dependencias externas. Además, ofrece una extensa de documentación disponible en la página web del proyecto y una lista de correo[58] muy activa, en la que los desarrolladores exponen sus problemas y sugerencias al resto de colegas. Por todos estos motivos, *Twitter4J* fue seleccionada para su uso en el proyecto frente al resto de opciones.

El tedioso proceso de realizar peticiones HTTP a la API de *Twitter* para recuperar la información en formato XML, es reemplazado en *Twitter4J* por el uso de simples llamadas a métodos de objetos *Java*. Las peticiones HTTP son gestionadas internamen-

te y toda la información recuperada es procesada y cargada en listas de objetos para facilitar su acceso y ordenación. Por otra parte, *Twitter4J* también es capaz de tratar las posibles excepciones en el proceso de recuperación de la información de *Twitter*, mediante mensajes de error.

El funcionamiento de *Twitter4J* puede dividirse en varias fases, que se ilustran en la siguiente figura:

Figura 12: Funcionamiento de Twitter4J

Twitter4J ha sido desarrollado y es periódicamente actualizado por el desarrollador japonés de alias *yusukey*[59]. Se trata de un proyecto de código abierto gratuito, distribuido con licencia *BSD*[60] y disponible para ser utilizado con fines tanto comerciales como no comerciales. Por otra parte, el proyecto admite donaciones económicas altruistas como forma de financiación extraordinaria.

La versión de *Twitter4J* utilizada en el proyecto ha sido la *2.0.10*.

4.3.4. MySQL

MySQL[61] es un sistema de gestión de base de datos relacional, multihilo y multiusuario propiedad de *Oracle Corporation* a raíz de la compra de *Sun Microsystems* en abril de 2009. Se ofrece bajo la *GNU GPL*[62] para cualquier uso compatible con esta licencia, pero aquellas empresas que quieran incorporarlo en productos privativos deben adquirir una licencia específica para tal fin.

En el caso particular del proyecto, el uso de la licencia libre estaba totalmente justificado. Por ello, y por la potencia y simplicidad que ofrece *MySQL*, se escogió como gestor de la base de datos para el almacenamiento en el entorno local.

4.3.5. Almacén de Datos de GAE

El almacén de datos es empleado por *GAE* para dotar a las aplicaciones de un entorno de almacenamiento escalable. Se ha diseñado para optimizar la lectura y la realización de consultas, y puede ejecutar varias operaciones en una única transacción recuperando la transacción entera si falla cualquiera de las operaciones. Esto resulta especialmente útil en el caso de las aplicaciones web distribuidas, en las que es posible que varios usuarios accedan a los mismos objetos de datos de forma concurrente.

A diferencia de las bases de datos tradicionales, el almacén de datos utiliza una arquitectura distribuida para gestionar el cambio de tamaño de los conjuntos de datos muy grandes. Una aplicación *App Engine* puede optimizar la forma en que se distribuyen los datos mediante la descripción de relaciones entre objetos de datos y la definición de índices para las consultas.

El almacén de datos de *App Engine* es totalmente consistente, pero ello no implica que se trate de una base de datos relacional. Aunque su interfaz tiene muchas de las funciones de las bases de datos tradicionales, las características únicas del almacén de datos suponen una forma diferente de diseñar y gestionar los datos. Por ejemplo,

las entidades del almacén de datos no tienen un esquema único: dos entidades del mismo tipo no están obligadas a contener las mismas propiedades o utilizar los mismos tipos de valores para las mismas propiedades. En consecuencia, la propia aplicación es la única responsable de garantizar que las entidades cumplan un determinado esquema cuando sea necesario.

4.3.6. Java Data Objects (JDO)

Google App Engine, a través del entorno de desarrollo *SDK Java*, incluye implementaciones de los *Objetos de Datos Java* (*JDO*) y de las *Interfaces del API de persistencia de Java* (*JPA*) para crear y persistir datos.

En el desarrollo del proyecto se ha optado por utilizar *Java Data Objects* para la persistencia de los datos. *JDO* ofrece varias ventajas con respecto a *JPA*, tal y como puede apreciarse en la siguiente tabla comparativa, y dispone de una mayor cantidad de documentación para su uso dentro de *GAE*. Ésta última, junto con el mayor grado de desarrollo en *GAE*, han sido las razones fundamentales por las que finalmente se seleccionó *JDO* en detrimento de *JPA*.

	Java Data Objects (JDO)	***Java Persistence API (JPA)***
Requisitos *SDK Java*	1.3 ó superior	1.5 ó superior
RDBM (Relational Data Base Modeling)	Soportado	Soportado
ORM (Object Role Modeling)	Soportado	No soportado
Catálogo de tipos de datos soportados	Grande	Mediano
DataNucleus	Soportado	No soportado
Grado de desarrollo en *GAE*	Alto	Medio-Bajo

Tabla 7: Comparativa entre Java Data Objects (JDO) y Java Persistence API (JPA)

4.3.7. Colas de Tareas de GAE

El *Sistema de Colas de Tareas* de *Google App Engine* permite ejecutar llamadas a URLs, que no hayan sido desencadenadas directamente por peticiones HTTP del cliente. De este modo es posible realizar cálculos en segundo plano, segmentados en pequeñas y discretas unidades llamadas tareas. La gestión de las colas es responsabilidad de *GAE*, y las peticiones almacenadas se van ejecutando en función de los atributos de la cola y de la disponibilidad de recursos del sistema.

Las colas de tareas son definidas en un fichero llamado *queue.xml*, en el que se especifica: su nombre, la periodicidad de ejecución de sus tareas y su "bucket-size" o tamaño. El uso de las colas permite crear aplicaciones con mayor capacidad de cálculo, mediante la segmentación en tareas cuya ejecución no alcance el límite de tiempo de cálculo impuesto por *GAE*. Sin embargo, las colas de tareas también están sometidas a sus propias restricciones como muestra la siguiente tabla:

Recurso	Restricciones
Tamaño de tarea individual	10 kilobytes
Número de colas activas (sin incluir cola por defecto)	10
Tasa de ejecución por cola	50 tareas por cola/segundo
Periodo de vida por tarea	30 días
Número de taras añadidas en un batch	100 tareas

Tabla 8: Restricciones de las colas de tareas de GAE

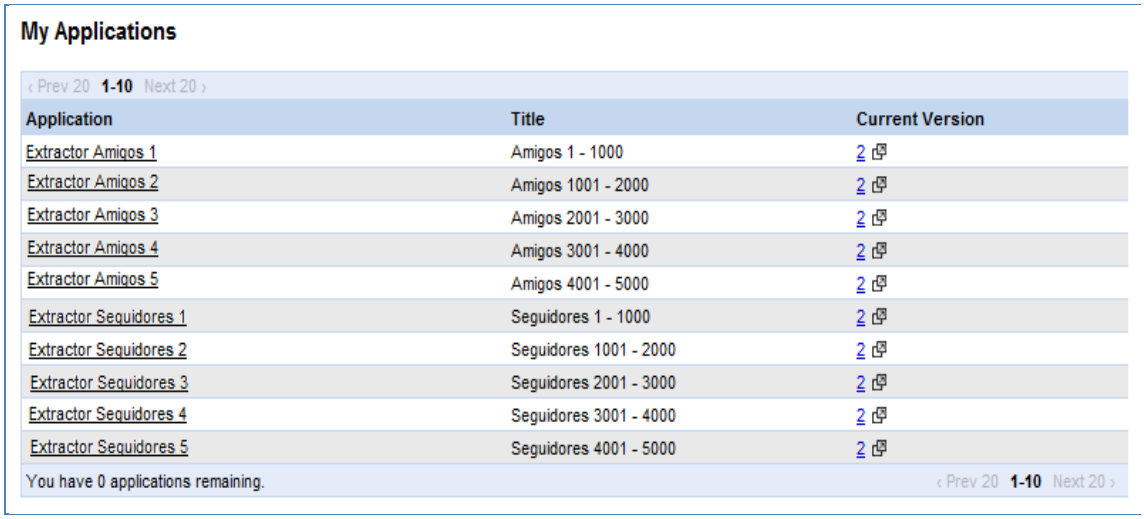
En el proyecto, las colas de tareas se han empleado para realizar el mayor número de peticiones de información al servicio de red social *Twitter* en segundo plano, sin saturar los recursos del sistema. En concreto, las aplicaciones en el entorno "cloud" (extractores de amigos y de seguidores) han utilizado una cola cada uno de periodicidad 20 m/s y un "bucket-size" de 5 unidades, para planificar las correspondientes peticiones de información de los nodos analizados.

4.4. Herramientas de Monitorización y Análisis

Además de las herramientas de desarrollo, en el proyecto se han empleado otras herramientas para la monitorización del sistema, y para el análisis y representación de la información obtenida. El seguimiento de las ejecuciones en el entorno "cloud" se ha llevado a cabo a través la *Consola de Administración* de *GAE*, y el estudio de los grafos sociales recuperados de *Twitter* mediante las herramientas *Graph-Tool* y *Pylab*.

4.4.1. Consola de Administración de GAE

La *Consola de Administración* de *Google App Engine* ofrece un acceso total a los parámetros de configuración de la aplicación y permite realizar un seguimiento de las ejecuciones o peticiones llevadas a cabo durante las últimas 24 horas. Para poder acceder a ella es necesario autenticarse desde el portal que ofrece *Google* y seleccionar alguna de las aplicaciones desarrolladas. En el caso particular del proyecto, tal y como puede observarse en la siguiente captura, las aplicaciones desarrolladas han sido 10.



My Applications

‹ Prev 20 **1-10** Next 20 ›

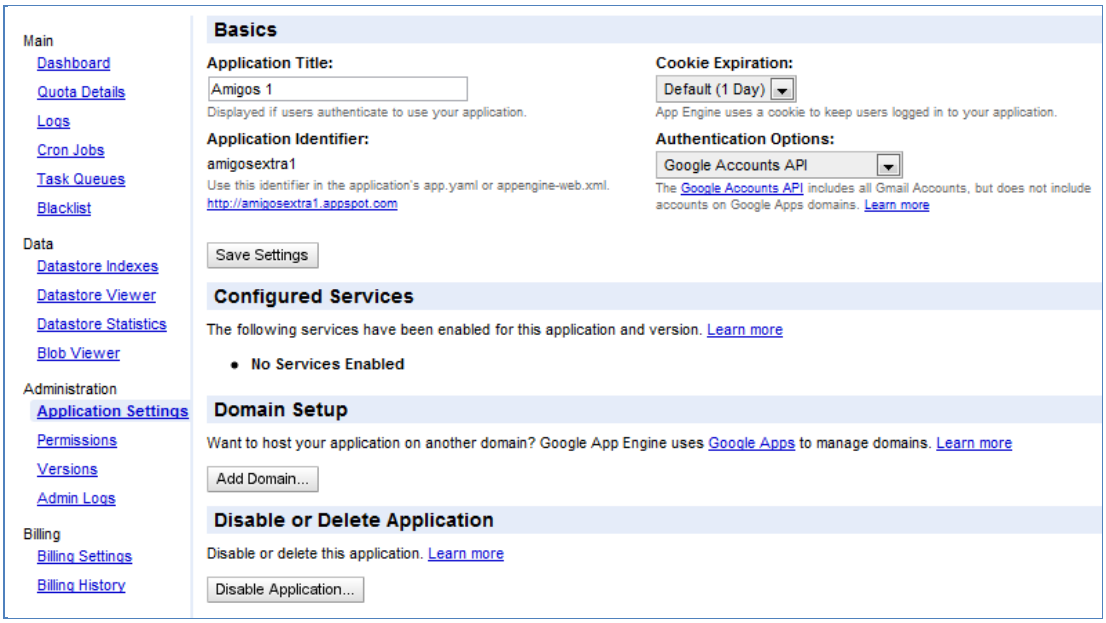
Application	Title	Current Version
Extractor Amigos 1	Amigos 1 - 1000	2
Extractor Amigos 2	Amigos 1001 - 2000	2
Extractor Amigos 3	Amigos 2001 - 3000	2
Extractor Amigos 4	Amigos 3001 - 4000	2
Extractor Amigos 5	Amigos 4001 - 5000	2
Extractor Seguidores 1	Seguidores 1 - 1000	2
Extractor Seguidores 2	Seguidores 1001 - 2000	2
Extractor Seguidores 3	Seguidores 2001 - 3000	2
Extractor Seguidores 4	Seguidores 3001 - 4000	2
Extractor Seguidores 5	Seguidores 4001 - 5000	2

You have 0 applications remaining. ‹ Prev 20 **1-10** Next 20 ›

Captura 3: Consola de Administración de GAE: listado de aplicaciones

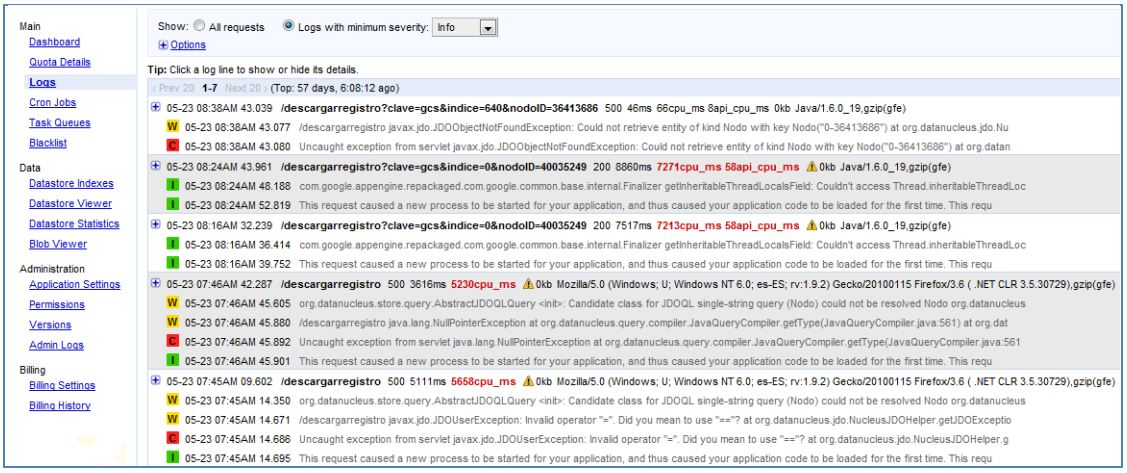
En concreto, desde la *Consola de Administración* es posible ejecutar las siguientes funcionalidades:

- **Configurar los parámetros de la aplicación**, incluyendo la posibilidad de establecer un subdominio *appspot.com* gratuito o un nombre de dominio de nivel superior.



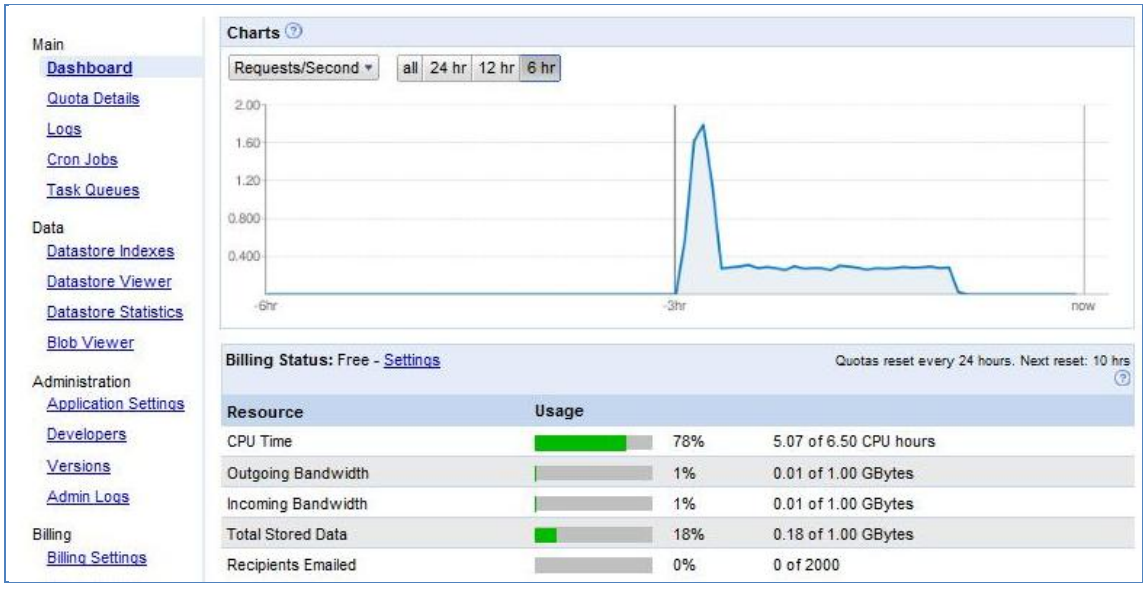
Captura 4: Consola de Administración de GAE - configuración de parámetros

- **Gestionar los permisos de la aplicación**, para permitir el acceso de varios desarrolladores, y **llevar un control de versiones de código**.
- **Probar nuevas versiones** de la aplicación **y cambiar la versión disponible para los usuarios** en cada momento.
- **Navegar por el almacén de datos de la aplicación**, administrar los índices y realizar consultas en lenguaje *GQL* sobre los datos almacenados.
- **Comprobar el estado de las tareas programadas** de la aplicación.
- **Visualizar los registros de errores y datos de acceso de las peticiones realizadas,** para analizar el tráfico durante un periodo de tiempo determinado.



Captura 5: Consola de Administración de GAE - registro de peticiones realizadas

- **Consultar gráficas y estadísticas acerca de los recursos** (almacenamiento, tiempo de CPU, número de peticiones, etc.) consumidos y disponibles por la aplicación.



Captura 6: Consola de Administración de GAE - gráficas y estadísticas de los recursos

Asimismo, la *Consola de Administración* de *GAE* se encuentra en estado constante de desarrollo de nuevas funcionalidades, que van siendo incorporadas con el paso del tiempo.

4.4.2. Graph-Tool

Graph-Tool[63] es un módulo de *Python* para la manipulación y el análisis estadístico de grafos. Entre otras funcionalidades permite: crear y editar grafos, recorrer sus vértices o aristas, calcular métricas estadísticas, ejecutar algoritmos topológicos, etc. Posee abundante documentación en su página web, incluyendo un tutorial con numerosos ejemplos.

A pesar de que gran parte de *Graph-Tool* está escrita en *Python*, sus estructuras de datos y algoritmos de cálculo han sido desarrollados en *C++* hacienda uso de la *Boost Graph Library* y técnicas de metaprogramación para optimizar el rendimiento. Este hecho ha sido determinante en la utilización del módulo para el proyecto, dado el gran tamaño de los grafos sociales extraídos de *Twitter* a analizar.

La versión de *Graph-Tool* utilizada en el proyecto ha sido la *2.2.5*.

4.4.3. PyLab

PyLab[64] es un paquete para el análisis numérico, la computación y la representación gráfica que agrupa los módulos de *Python*: *NumPy*, *SciPy*, *Matplotlib*, y *IPython*. A pesar de amplio catálogo de funcionalidades que presenta, la utilización de *Pylab* en el proyecto se ha reducido únicamente a la generación de gráficas a partir de datos calculados con *Graph-Tool* y los extraídos de forma directa en el seguimiento de usuarios.

5. Proceso de Extracción, Seguimiento y Análisis

En este apartado, se procede a realizar una descripción pormenorizada de cada una de las fases que componen el proceso de extracción, seguimiento y análisis de la información del servicio de red social *Twitter*, mencionando primeramente las limitaciones de la fase de extracción, según su naturaleza. Todo el proceso puede descomponerse en una serie de fases o tareas que se han llevado a cabo de forma aislada o periódica, en las que se han utilizado las herramientas de desarrollo, monitorización y análisis descritas en el apartado anterior.

5.1. Limitaciones del Proceso de Extracción

El proceso de extracción de la información del servicio de red social *Twitter* mediante *Google App Engine*, está sujeto a ciertas limitaciones impuestas tanto por la API de *Twitter*, como por el uso de las cuentas gratuitas de *GAE*. De hecho, uno de los principales objetivos del proyecto es estudiar las prestaciones de las aplicaciones alojadas sobre este tipo de infraestructura de forma gratuita. Por lo tanto, las limitaciones que afectan a todo el proceso se clasifican en dos grupos: las restricciones determinadas por la API de *Twitter* y las derivadas del uso de cuentas gratuitas.

5.1.1. Restricciones de la API de Twitter

Las extracciones de información de *Twitter*, como no podía ser de otra manera dada la gran cantidad de información requerida, se han realizado mediante peticiones a su API. En particular, la librería *Java Twitter4J* ha actuado como "binding" para tal efecto, y se ha ocupado del análisis sintáctico del código HTML obtenido, simplificando enormemente el trabajo.

La API[43] de *Twitter* es relativamente potente y extensa, además de estar abundantemente documentada. Sin embargo, *Twitter* impone una serie de condicio-

nes y restricciones sobre su uso, que le permiten llevar un control de los recursos para evitar caídas y saturaciones del sistema. Dichas restricciones afectan a los dos módulos basados en el paradigma REST, que dividen la API por motivos históricos: *REST API* y *Search API*. Las restricciones más importantes de los módulos citados pueden observarse en la siguiente tabla:

Módulo	Restricciones
Común	1.000 actualizaciones/día desde cualquier dispositivo
Común	250 mensajes directos/día desde cualquier dispositivo
Común	2.000 seguidores máximo
REST API	150 peticiones/hora por usuario autenticado
REST API	**150 peticiones/hora por IP no "whitelisted"**
REST API	20.000 peticiones/hora por IP "whitelisted" (previa aprobación)
Search API	No proporcionado, pero >> 150 peticiones/hora por IP

Tabla 9: Restricciones de la API de Twitter

Para llevar a cabo las extracciones de información se han realizado peticiones mediante el paradigma REST al módulo de información de usuarios de *Twitter*, y por consiguiente, todo el proceso ha estado sometido a un cuello de botella derivado de las restricciones de dicho módulo, el *REST API*. Tales restricciones varían profundamente en función de si las peticiones se realizan desde una dirección IP incluida en la "whitelist" de *Twitter*, o no. Ésta lista incluye una serie de direcciones privilegiadas, que pueden disfrutar de ciertos servicios en condiciones más ventajosas. El procedimiento para incluir una dirección IP dentro de la "whitelist" de *Twitter* consiste en rellenar una solicitud, que es enviada al departamento técnico de *Twitter*, estudiada y aprobada o no en función del uso que vaya hacerse de la dirección.

Teniendo en cuenta el gran número de usuarios a estudiar pertenecientes al conjunto raíz, las posibilidades para poder realizar todas las peticiones diarias necesa-

rias eran dos: incluir una dirección IP dentro de la "whitelist" de *Twitter* para disfrutar de unas restricciones menos estrictas, o llevar a cabo las peticiones desde un conjunto considerable de distintas direcciones IP. Dado que el proyecto se concibió desde un principio como un estudio mixto sobre los servicios de redes sociales y el paradigma "cloud computing", se optó por la solución de emplear distintas direcciones IP para las peticiones desde la nube, tal y como se describe en el subapartado 5.2.

5.1.2. Restricciones de las Cuentas Gratuitas de GAE

La plataforma *Google App Engine* permite ejecutar aplicaciones web en la infraestructura *Google* de forma sencilla y eficiente. *GAE* ofrece una serie de servicios y recursos, de los que se puede disfrutar hasta un límite predeterminado con una cuenta gratuita. Algunos de los recursos permiten sobrepasar dicho límite gratuito habilitando la cuota de facturación, lo que conlleva el abono de un importe de acuerdo a unas tarifas fijas. Otros, sin embargo, son más estrictos y no pueden sobrepasarse bajo ningún concepto, como por ejemplo el límite de 30 segundos de ejecución por tarea individual. Partiendo de la base de que el proyecto no incluye gastos derivados del uso de *GAE*, no se han habilitado las cuotas de facturación de las aplicaciones y por consiguiente solo se ha hecho uso de los recursos gratuitos.

En la Tabla 10 pueden observarse las restricciones más importantes de las cuentas gratuitas predeterminadas de *GAE*, incluyendo los límites por día y por minuto para los recursos cuya cuota es renovada diariamente por *Google*.

Recurso	**Restricciones por Aplicación**	
Número de aplicaciones	10 por usuario (cuenta *Google*)	
Tiempo ejecución tarea individual	**30 segundos**	
Peticiones	1.300.000 unds/día	7.400 unds/min
Ancho de banda (salida)	10 GB/día	56 MB/min

Ancho de banda (entrada)	10 GB/día	56 MB/ min
Tiempo de CPU	**6,5 horas CPU/día**	**15 min CPU/min**
Llamadas API almacenamiento	10.000.000 unds/día	57.000 unds/min
Datos almacenados	**1 GB**	
Datos enviados API almacenamiento	12 GB/día	68 MB/min
Datos recibidos API almacenamiento	115 GB/día	659 MB/min
Tiempo de CPU almacén de datos	46 horas CPU/día	15 min CPU/min
Tareas	100.000 tareas/día	
Deployments	1.000 unds/día	

Tabla 10: Restricciones de las cuentas gratuitas de GAE

Las restricciones de las cuentas gratuitas sobre el tiempo de CPU y el espacio de almacenamiento, han determinado la replicación de las aplicaciones de extracción en el entorno "cloud" (véase el subapartado 3.3), y el tamaño del conjunto raíz o conjunto de usuarios a explorar, tal y como se especifica en el siguiente subapartado 5.2.1. Sin embargo, el principal problema encontrado con *GAE* no es una consecuencia del uso de las cuentas gratuitas, si no que viene dado por el tiempo máximo de ejecución de las tareas individuales, idéntico para las cuentas con facturación habilitada. Este tiempo nunca ha de superar los 30 segundos. Para poder afrontar este problema las tareas se han segmentado en subtareas más pequeñas, y se han gestionado mediante el *Sistema de Colas* de *GAE*. Asimismo, el número de amigos y de seguidores extraídos de cada usuario se ha limitado a 250 como máximo (véase el subapartado 6.1).

5.2. Fases del Proceso

El proceso de extracción, seguimiento y análisis de la información del servicio de red social *Twitter* mediante *Google App Engine* se realiza en varias fases, algunas de las cuales se llevan a cabo en el entorno local y otras en la infraestructura "cloud". El siguiente diagrama muestra la secuencia de ejecución de dichas fases:

Figura 13: Diagrama de las fases del proceso de extracción, seguimiento y análisis

A continuación, se describen en detalle cada una de las fases que conforman todo el proceso: desde la generación del conjunto raíz, hasta el análisis de la información del servicio de red social *Twitter*, pasando por su extracción en el entorno "cloud" y su posterior recuperación para el entorno local.

5.2.1. Generación del Conjunto Raíz

Para llevar a cabo las extracciones periódicas de información del servicio de red social *Twitter*, se ha seleccionado un conjunto de usuarios sobre el que iniciar el proceso, el conjunto raíz. El tamaño de dicho conjunto es una consecuencia directa de las restricciones impuestas por las cuentas gratuitas de *GAE*. Estas, como se ha detallado anteriormente, establecen un límite de consumo de 6,5 horas de CPU por día, o lo que es lo mismo por periodo de extracción. Además, debe tenerse en cuenta que se ha fijado un límite de 250 amigos y 250 seguidores como máximo por usuario (véase subapartado 6.1), y que se dispone de un total 10 aplicaciones desplegadas en la infraestructura "cloud" (5 extractores de amigos y 5 extractores de seguidores).

Tras un periodo de experimentación y prueba con todas estas variables, se determinó asignar un conjunto de 1.000 usuarios para cada uno de los extractores. De esta forma, éstos efectuaban un consumo aproximado del 70% del tiempo de CPU diario por cada periodo de extracción, dejando libre un margen suficiente para contemplar posibles escenarios de saturación de *GAE* o de la API de *Twitter*. Por consiguiente, y teniendo en cuenta que cada conjunto de 1.000 usuarios requiere un extractor de amigos y otro de seguidores, se alcanza un tamaño de conjunto raíz de 5.000 usuarios (5 extractores x 1.000 usuarios/extractor = 5.000 usuarios).

Una vez determinado el tamaño del conjunto raíz, para seleccionar a sus miembros se comenzó añadiendo un usuario semilla, a partir del cual se fueron incluyendo amigos y seguidores mediante una búsqueda en anchura. El usuario semilla escogido fue *guillermocbns*[65] de ID *Twitter* 40035249, con 62 amigos y 20 seguidores. En la Figura 14 puede apreciarse una pequeña muestra del desarrollo del conjunto raíz a partir de la semilla *guillermocbns*. Los usuarios añadidos incluyen el número de amigos y seguidores que han aportado al conjunto, y en algunos casos su localización, dependiendo si poseen o no habilitada la opción "geolocation" de *Twitter*.

Figura 14: Muestra de desarrollo del conjunto raíz

Dado que el objetivo principal era obtener un grafo conexo de relaciones (amigos-seguidores) lo más extenso posible en cuanto a la profundidad de las relaciones, pero conservando las propiedades de los usuarios populares, se limitó la inclusión en el conjunto raíz a 250 amigos y 250 seguidores máximo por usuario ya añadido. De esta forma, se evitó generar un grafo con aspecto de sistema solar, con excesivos usuarios relacionados con otro muy popular, actuando a modo de sol. El análisis de un grafo de este tipo hubiera aportado una idea errónea acerca del comportamiento real de los usuarios de *Twitter* y unas conclusiones poco extrapolables a situaciones fuera del conjunto raíz. Asimismo, y para optimizar el proceso de composición del conjunto raíz, en el caso de que se tratara de añadir un usuario ya incluido, éste se contabilizaba como amigo o seguidor para el cálculo de los límites por usuario (250 amigos y 250 seguidores), pero se omitía su reinserción.

A pesar de disponer de un conjunto raíz compuesto por 5.000 usuarios, un número relativamente pequeño de ellos se trataba de usuarios privados que no permitan el acceso a su información. Sin embargo, estos usuarios no fueron excluidos del conjunto durante la fase de generación ya que su filtrado resultaba altamente costoso. Todos los usuarios pueden variar en todo momento su configuración de privacidad, y por lo tanto pierde sentido hacer éste tipo de distinciones como si se tratara de una propiedad estática y no dinámica. Por consiguiente, se contempló el hecho de que usuarios de perfil privado pudieran pasar a ser usuarios públicos durante el periodo de extracción, y viceversa.

5.2.2. Lanzamiento de las Peticiones de Extracción

Dada la arquitectura mixta del sistema, formada por módulos locales y otros en el entorno “cloud”, existen varias fases del proceso en las que ambos entornos interactúan. Una de dichas fases es la del lanzamiento de peticiones de extracción, en la que desde del entorno local, un cliente (programa *Java*) realiza peticiones HTTP a una serie de servidores (los extractores), que se encuentran desplegados en la infraestruc-

tura "cloud". Dichas peticiones son atendidas por "servlets" alojados en *GAE*, que se ocupan de la ejecución de tareas previamente programadas.

El objetivo de esta fase es desencadenar diariamente durante dos semanas, las extracciones de los amigos y seguidores de los usuarios del conjunto raíz, por las aplicaciones desplegadas en el entorno "cloud". Para ello se realizan secuencialmente desde el entorno local, 10.000 peticiones (5.000 para extraer amigos + 5.000 para extraer seguidores) cada día y a una hora similar, agrupadas de 1.000 en 1.000. Cada grupo va destinado a un servidor-extractor diferente que se encarga de procesar las peticiones, y de llevar a cabo las tareas correspondientes tal y como se explica en el siguiente apartado. La distribución de las 10.000 peticiones de extracción de amigos y seguidores de los usuarios del conjunto raíz, se realiza de acuerdo a un índice.

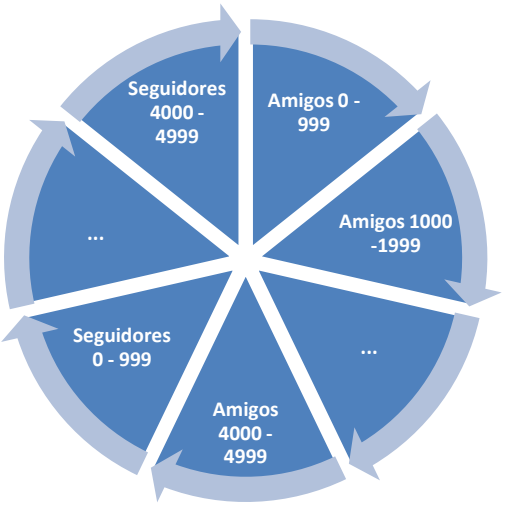


Figura 15: Distribución de las peticiones de extracción

Cada una de las peticiones incluye una serie de parámetros, que son interpretados por los servidores-extractores para determinar la tarea a realizar. El formato exacto de las peticiones HTTP de extracción es el siguiente:

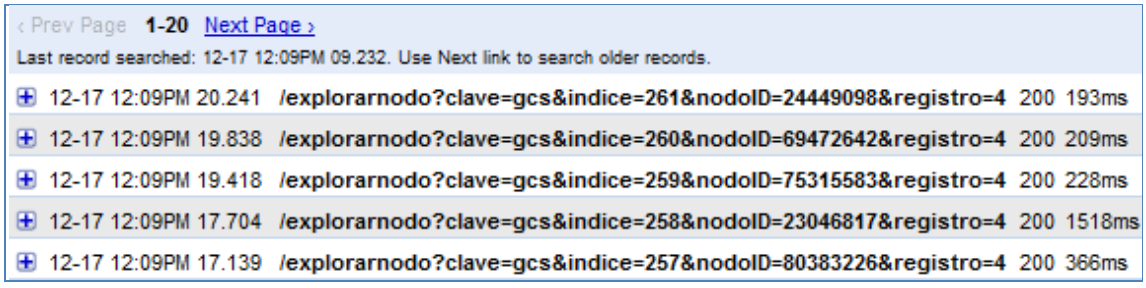
```
http://www.nombreServidor.appspot.com/explorarnodo?clave
    =XXXXX &indice=XXXXX&nodoID=XXXXX&registro=XXXXX
```

`nombreServidor`	Nombre del servidor de entre los 10 existentes en formato URL
`explorarnodo`	Subruta del "servlet" encargada del procesamiento de las peticiones de extracción
`clave`	Clave a modo de contraseña para evitar bots e indexadores
`indice`	Posición del usuario en la BBDD local del conjunto raíz
`nodoID`	Identificador *Twitter* del usuario
`registro`	Identificador del periodo de extracción (0-14)

Tabla 11: Formato de las peticiones de extracción

La riqueza en parámetros de las peticiones HTTP, permite no solo identificar a cada servidor-extractor la tarea a realizar, sino que además facilita el seguimiento del proceso mediante el *Monitor de Aplicaciones* de *GAE*. Por su parte, la utilidad del parámetro clave se limita únicamente a filtrar las peticiones realizadas por bots e indexadores, como *Googlebot*. Dichas peticiones podrían desencadenar ejecuciones de código no deseadas, ni registradas desde el entorno local, pero en ningún caso se trata de un mecanismo de seguridad de acceso y por lo tanto no debe ser tratado como tal.

En la Captura 7 puede observarse el modo en el que se recogen las peticiones de extracción, a través de la ventana de logs del *Monitor de Aplicaciones*:



Captura 7: Logs de las peticiones de extracción

Por otra parte, y teniendo en cuenta los errores de conexión o distintas excepciones que pueden producirse durante el largo proceso del lanzamiento de peticiones,

todas aquellas peticiones no realizadas correctamente son recogidas en un archivo de error. De esta forma, una vez concluido el proceso pueden consultarse y lanzarse de nuevo de forma manual a través del navegador web *Mozilla Firefox*.

La situación del lanzador de peticiones en el entorno local y no en la infraestructura "cloud", permite disponer de una solución fácilmente adaptable a unas variables de ejecución diferentes, como por ejemplo un conjunto raíz mayor o un número distinto de servidores de extracción. Asimismo, la distribución de las peticiones puede alterarse libremente modificando unos parámetros, lo que permite asignar una carga de trabajo diferente a cada uno de los servidores.

5.2.3. Extracción de los Grafos Sociales

Cada día y durante dos semanas, una vez las peticiones de extracción hayan sido enviadas desde el entorno local a la infraestructura "cloud", llega el momento de realizar las extracciones de amigos y seguidores. Tal y como se recoge en el apartado anterior, cada servidor recibe 1.000 peticiones con unos parámetros que definen la extracción: índice, identificador *Twitter* del usuario, registro, etc. En todos ellos se lleva a cabo un proceso idéntico, excepto por los usuarios asignados, y porque se procede a extraer los amigos o los seguidores, en función del tipo del servidor: 5 extractores de amigos y 5 extractores de seguidores.

Cada petición es atendida por un "servlet" del tipo `ExplorarNodo`. Dicho "servlet" realiza una primera comprobación sobre si ya se ha realizado la extracción diaria o por periodo de registro para ese usuario, o no. Esto es posible dado que los objetos persistentes que almacenan las extracciones en el entorno "cloud", poseen un identificador único o clave primaria formada por los parámetros de la petición: "`registro-nodoID`". Únicamente en el caso de que no se haya producido ya dicha extracción, se crea una tarea que es añadida a la cola de tareas de *GAE* `extraeramigos` o `extraerseguidores`, según corresponda.

Dichas tareas se traducen en nuevas peticiones HTTP, para los "servlets" `ExplorarAmigos` y `ExplorarSeguidores` respectivamente. Cuando un "servlet" de estos dos tipos recibe una petición, recupera los parámetros de la extracción y tras un proceso de autenticación, demanda a la API de *Twitter* los amigos o los seguidores del usuario para almacenarlos. En la siguiente captura pueden apreciarse algunos logs de tareas de extracción visualizados a través del *Monitor de Aplicaciones* de *GAE*:

Captura 8: Logs de las tareas de extracción

La comunicación con la API de *Twitter* se lleva a cabo mediante la librería *Java Twitter4J*, que facilita enormemente el trabajo. Dicha comunicación está sujeta a cuatro posibles tipos de error representados como códigos RFC[66] del protocolo HTTP, antes los cuales las aplicaciones de extracción reaccionan de diferente manera.

Código	Error	Resolución
400 (Bad Request)	Límite de la API por dirección IP alcanzado	La tarea vuelve a añadirse a la cola, para tratar de ejecutarse desde otra dirección IP
401 (Unauthorized)	Información restringida	La tarea se desecha, se trata de un usuario privado
502 (Bad Gateway)	*Twitter* está caído	La tarea vuelve a añadirse a la cola, a la espera de la restitución del servicio
503 (Service Unavailable)	*Twitter* está saturado	La tarea vuelve a añadirse a la cola, a la espera de la restitución del servicio

Tabla 12: Errores de la API de Twitter como códigos RFC del protocolo HTTP

Por lo tanto, cada tarea se pospone indefinidamente hasta que se completa de forma satisfactoria o se desecha por requerir información inaccesible de un usuario privado. Las características de *GAE* permiten que al ejecutarse un buen número de tareas en paralelo, se disponga de un rango considerable de diferentes direcciones IP, que permiten numerosas peticiones a la API de *Twitter* sin excesivas denegaciones de servicio por límite de cuota alcanzado.

A la hora de almacenar los amigos o seguidores de un usuario con la limitación de los 250 elementos como máximo, se guarda también el número de amigos o seguidores totales en *Twitter*. En los usuarios con menos de 5.000 amigos o seguidores esta acción es algo trivial, pues el número equivale al tamaño del listado devuelto por los métodos `getFriendsIDs(…)` y `getFollowersIDs(…)` de *Twitter4J*. Por el contrario, para los otros casos es necesario realizar una petición adicional mediante el método `showUser(…)` de *Twitter4J*.

5.2.4. Seguimiento de Usuarios

Además de las extracciones diarias de los amigos y seguidores, se ha programado un seguimiento a dos usuarios particulares, para conocer sus "tweets" y la influencia de éstos sobre los "tweets" de sus seguidores.

En la elección de los dos usuarios a seguir, se ha optado por tomar uno con un número de amigos y seguidores relativamente pequeño, y otro con un número mayor. De esta forma, y teniendo ambos un grado de participación activo en *Twitter*, es posible estudiar la influencia de dos tipos de usuarios existentes en la red. Por otra parte, los dos se han seleccionado de entre los amigos y seguidores del usuario *guillermocbns*, para mantener un grado de proximidad elevado que facilitara el proceso. El usuario "menos popular" estudiado es *zaquito*[67], mientras que el "más popular" es *googlappengine*[68]. En la Tabla 13 pueden consultarse más detalles acerca de ambos usuarios.

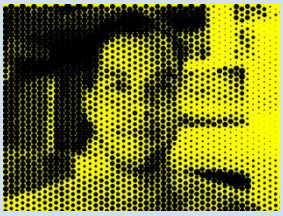
	Usuario "menos popular"	**Usuario "más popular"**
Nombre	*zaquito*	*googlappengine*
ID *Twitter*	52360686	60959947
Antigüedad	Tue Jun 30 12:27:08 CEST 2009	Tue Jul 28 19:08:28 CEST 2009
Localización	Madrid	Mountain View
Imagen de perfil *	 **Figura 16 Imagen de perfil del usuario zaquito**	 **Figura 17: Imagen de perfil del usuario googlappengine**
Núm. amigos *	53	200
Núm. seguidores *	59	219
Núm. favoritos *	39	0
Núm. listas *	2	34
Naturaleza	Estudiante de Ingeniería Informática de la UC3M	Canal de noticias y comentarios relacionados con *GAE*
Grado Participación	Activo	Activo
*** NOTA: información vigente a día 31/12/09**		

Tabla 13: Detalles de los usuarios del seguimiento

Para el seguimiento de los usuarios, mucho más sencillo que las extracciones de amigos y seguidores, no se han utilizado recursos en el entorno "cloud". Todo el proceso se ha llevado a cabo en el entorno local, para lo que se han empleado peticiones de información a la API de *Twitter* mediante llamadas a métodos de *Twitter4J*. Los resultados obtenidos se han almacenado en ficheros planos para su posterior análisis.

5.2.5. Lanzamiento de las Peticiones de Recuperación

De un modo similar al empleado para el lanzamiento de las peticiones de extracción, se lleva a cabo el de las peticiones de recuperación de los datos. El objetivo de esta fase es, en términos generales, recuperar para el entorno local toda la información extraída de *Twitter* y almacenada en la infraestructura "cloud" por las distintas aplicaciones o servidores de extracción. Para ello han de realizarse 10.000 peticiones (5.000 para datos de amigos + 5.000 para datos de seguidores) por cada periodo de extracción. La distribución de las 10.000 peticiones de recuperación de datos por periodo de extracción se realiza de forma idéntica a como se hace para las peticiones de extracción. Igualmente, esta distribución puede variarse con algún objetivo concreto.

Cada una de las peticiones de recuperación es atendida por un "servlet" del tipo `descargarNodoGraphml`, e incluye una serie de parámetros que son interpretados para determinar los datos que desean ser recuperados. En función de dichos parámetros, el servidor realiza una consulta de los datos requeridos en su base de datos del entorno "cloud" y los muestra como el contenido de una `HttpServletResponse`. En ese momento, el cliente HTTP en local recupera los datos y los guarda en un fichero donde se irán almacenando todos los datos recuperados por periodo de extracción. El formato de las peticiones HTTP de recuperación es el siguiente:

`http://www.nombreServidor.appspot.com/descargarNodoGraphml?clave=XXXXX&indice=XXXXX&nodoID=XXXXX&registro=XXXXX`	
`nombreServidor`	Nombre del servidor de entre los 10 existentes en formato URL
`descargarNodoGraphml`	Subruta del "servlet" encargada del procesamiento de las peticiones de recuperación de datos
`clave`	Clave a modo de contraseña para evitar bots e indexadores
`indice`	Posición del usuario en la BBDD local del conjunto raíz
`nodoID`	Identificador *Twitter* del usuario

`registro`	Identificador del periodo de extracción (0-14)

Tabla 14: Formato de las peticiones de recuperación de datos

En la siguiente captura, puede observarse el modo en el que se recogen las peticiones de recuperación de datos, a través de la ventana de logs del *Monitor de Aplicaciones* de *GAE*:

Captura 9: Logs de las peticiones de recuperación de datos

Por otra parte, los errores de conexión o distintas excepciones son recogidos con el mismo propósito que durante el lanzamiento de las peticiones de extracción, y las peticiones de recuperación de datos también incluyen un parámetro clave que actúa como mecanismo de control de accesos indeseados.

Esta fase puede ejecutarse en cualquier momento del proceso, siempre y cuando se trate de recuperar información que ya se encuentre disponible en el entorno "cloud". Es decir, extracciones de amigos y seguidores que ya se hayan completado, y por lo tanto tengan un periodo de registro inferior al del día actual. Por otra parte, para evitar alcanzar los límites de las cuentas gratuitas de *GAE*, debe tenerse en cuenta que la recuperación de los amigos o seguidores de los 1.000 usuarios asociados a un servidor-extractor por periodo de registro, conlleva aproximadamente un 60% de la cuota diaria de tiempo de CPU. Por ello, las fases de lanzamiento de peticiones de recuperación de datos se han programado en el tiempo, si bien no de una forma tan estricta como las de peticiones de extracción, puesto que no se requería una planificación diaria como con las peticiones de extracción.

5.2.6. Análisis de la Información

La última fase del proceso corresponde al análisis de la información extraída del servicio de red social *Twitter*. Para que dicha fase pueda llevarse a cabo es necesario que toda la información se encuentre disponible en el entorno local, dado que *GAE* no proporciona suficiente potencia de cálculo mediante cuotas gratuitas para tal efecto. Esto equivale a que todas las fases previas se hayan desarrollado sin problemas y en particular, que las peticiones de recuperación para cada uno de los periodos de extracción hayan descargado correctamente la información almacenada en la infraestructura "cloud". En cuanto a los datos del seguimiento de usuarios, según el proceso se encuentran por defecto disponibles en el entorno del análisis. En ambos casos, la información es contenida en ficheros planos: formato XML para la información extraída del servicio de red social *Twitter* y texto para los datos de seguimiento de usuarios.

En el análisis de la información pueden distinguirse dos procedimientos, en función de la naturaleza de los datos: los grafos sociales o el seguimiento de usuarios. Para el análisis de los grafos sociales se ha empleado el módulo de *Python* para la manipulación y el análisis estadístico de grafos *Graph-Tool*[63] y el paquete de computación numérica y representación gráfica *PyLab*[64]. En concreto, los métodos y gráficas empleados para cada uno de los estudios han sido los siguientes:

Estudio	Métodos *Graph-Tool*	Gráficas *Pylab*
Estadísticas Generales	`num_vertices(…)`, `num_edges(…)`, `vertex_average(…)`, `edge_average(…)`	-
Análisis del Grado Nodal	`vertex_hist(…)`	`errorbar`
Análisis del *Clustering Coefficient*	`local_clustering(…)`, `global_clustering(…)`	`scatter`
Análisis Evolutivo	`out_edges(…)`, `in_edges(…)`	`scatter`

Tabla 15: Métodos Graph-Tool y gráficas PyLab empleados por estudio de análisis

Por su parte, para el análisis de los datos relativos al seguimiento de usuarios se ha utilizado un pequeño programa desarrollado en *Java*. La utilidad de dicho programa se reduce a contabilizar el número de apariciones de los "topics" mencionados por los usuarios *zaquito* y *googlappengine* entre los "tweets" de sus seguidores. El resto de datos (número de "retweets", de respuestas y de menciones) se han extraído directamente del servicio de red social Twitter, y todos ellos se han representado en gráficas haciendo uso de *Python* y del paquete *PyLab*.

6. Resultados del Análisis

En el siguiente apartado, primeramente se presentan las consideraciones generales que deben ser tenidas en cuenta para la correcta interpretación de los resultados del análisis. A continuación son enunciados y explicados los resultados obtenidos tanto para el análisis de los grafos sociales, como para el análisis del seguimiento de usuarios.

6.1. Consideraciones Generales

Para poder interpretar correctamente los resultados obtenidos durante el proyecto que nos ocupa, es necesario tomar en consideración una serie de puntos fundamentales de los que parte el proceso. Dichos puntos se refieren tanto a parámetros del mecanismo de extracción, como a asunciones durante la fase de análisis de la información.

Tomando la red social *Twitter* como un inmenso grafo social compuesto por usuarios representados como vértices, en ella pueden distinguirse dos tipos de relaciones o aristas entre los nodos: los amigos (“following”) y los seguidores (“followers”). Los dos tipos de relaciones son unidireccionales, a pesar de que la existencia de una relación de un tipo entre dos vértices en una dirección determinada, implique una relación de otro tipo en la dirección opuesta. Es decir, si un usuario A tiene una relación de seguimiento con otro usuario B, forzosamente B tendrá una relación de amistad con A. De igual manera, la desaparición de alguna de las relaciones conlleva la desaparición de su par.

En la siguiente figura, se ilustra un resumen de la explicación para el ejemplo de dos usuarios o vértices conocidos como A y B:

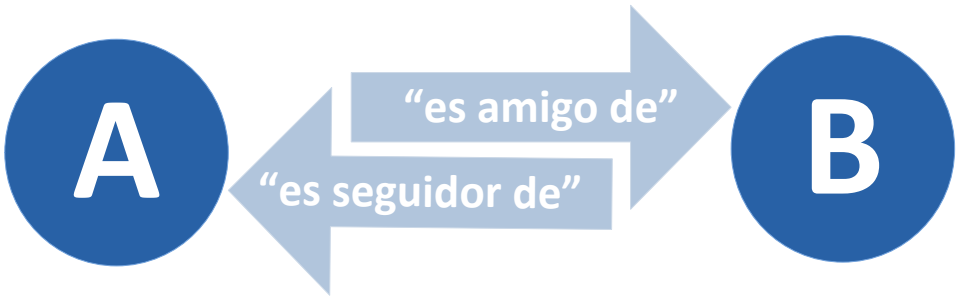


Figura 18: Relaciones entre usuarios en Twitter

En el proyecto, para poder distinguir el desigual comportamiento de los nodos en función del tipo de relación que exista entre ellos, se han considerado dos grafos diferentes: el grafo de amigos y el grafo de seguidores (dicha distinción también se ha adoptado en otros estudios similares[69] acerca de *Twitter*). Consecuentemente, ambos grafos contendrán únicamente relaciones o aristas unidireccionales de un solo tipo entre sus vértices: el grafo de amigos solo dispondrá de relaciones de amistad, mientras que el grafo de seguidores hará lo propio con las relaciones de seguimiento. Sin embargo, tal y como se especifica en el subapartado 5.2.1, los dos grafos se han construido a partir del mismo conjunto raíz de 5.000 usuarios. La única diferencia en el proceso de generación estriba en que para cada uno de los usuarios raíz han sido extraídos sus 250 primeros amigos o seguidores, en función del grafo a construir.

Se han considerado 250 como el máximo número de amigos o seguidores iniciales para cualquier nodo de ambos grafos. Si bien esta cantidad puede estar muy alejada del número real para ciertos usuarios muy populares, ha sido necesario tomar un valor máximo para poder acotar el conjunto de información a analizar dadas las limitaciones de las cuentas gratuitas de *GAE*. El valor de 250 relaciones se ha mostrado adecuado en las pruebas preliminares del proyecto para por un lado, poder recopilar información de un número de nodos lo suficientemente grande como para obtener conclusiones fiables y por otro, reflejar la singularidad de los usuarios muy populares con un valor de 250 amigos o seguidores cercano al infinito para la muestra del estudio. En cualquier caso, esta restricción solo aplica en el periodo de extracción inicial y

puede verse superada por conexiones internas entre nodos a partir de ese momento, tal y como se demuestra en las gráficas del subapartado 6.2.2. En otras palabras, a pesar de que inicialmente el número de amigos o seguidores totales de un usuario se limite a 250 unidades para el grafo extraído, dicho usuario podrá sumar nuevas relaciones durante el calendario de extracción con otros nodos que se encontrarán inicialmente dentro de la muestra de estudio.

En la siguiente figura, se representa de forma gráfica la limitación de 250 relaciones por usuario para el periodo de extracción inicial. En un primer momento se observa como el usuario o nodo A posee únicamente 250 relaciones registradas dentro de la muestra de estudio, a pesar que en el conjunto total de la red *Twitter* está relacionado con otros usuarios como es el caso del nodo C. Sin embargo, en periodos futuros del calendario de extracción, A establece nuevas relaciones con nodos que sí estaban presentes en la muestra inicial del estudio, es el caso del nodo B. En ésta ocasión, dichas relaciones sí son almacenadas sin que aplique la limitación de 250 unidades.

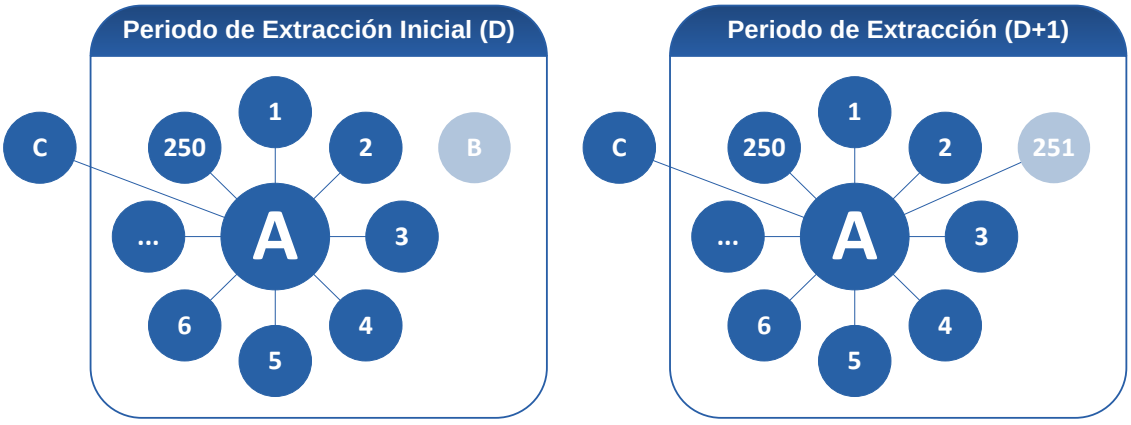


Figura 19: Limitación inicial de 250 relaciones por usuario

Por otra parte, en el análisis evolutivo de los grafos extraídos se ha empleado una ventana temporal de 2 semanas (14 días) desde el 13/12/2009 hasta el 26/12/2009. Es decir, la misma información de los grafos sociales se ha extraído diariamente durante dicho periodo de tiempo, para estudiar la evolución de su compor-

tamiento. En el caso del seguimiento de usuarios la ventana temporal ha tenido una longitud de 4 semanas (28 días), desde el 13/12/2009 hasta el 09/01/2010. En el subapartado 7.1.2 puede consultarse en mayor profundidad el calendario de extracción.

6.2. Análisis de los Grafos Sociales

La información sobre los grafos sociales extraída del servicio de red social *Twitter* ha sido analizada en base a diferentes métricas, y se han obtenido una serie de resultados que son expuestos en los siguientes subapartados. Como se ha mencionado con anterioridad, se han distinguido dos tipos de grafos sociales: el grafo de amigos con relaciones de amistad y el grafo de seguidores con relaciones de seguimiento. Todos los análisis realizados y descritos en los siguientes subapartados, se han realizado sobre ambos grafos para poder distinguir y comparar sus resultados.

6.2.1. Estadísticas Generales

Durante las 2 semanas (14 días) de duración del periodo de extracción para los grafos sociales se han registrado las estadísticas generales de los grafos de amigos y seguidores. Para cada grafo y periodo se ha anotado el número de vértices o usuarios y el número de aristas internas o relaciones de amistad-seguimiento existentes entre ellos. A partir de dichos datos se ha calculado la variación diaria porcentual de cada uno de los valores, la media de relaciones por usuario y se han hallado los valores promedio. La Tabla 16 muestra los valores promedio de los atributos de los grafos de amigos y los grafos de seguidores, durante el periodo de extracción:

	Vértices (Usuarios)	Δ% Vértices (Usuarios)	Aristas (Relaciones)	Δ% Aristas (Relaciones)	Aristas / Vértice
Amigos	339.600	-0,09%	596.121	-0,10%	3,51
Seguidores	351.012	0,09%	570.407	0,05%	3,25

Tabla 16: Estadísticas generales de los grafos de amigos y seguidores

Los siguientes gráficos (Gráfico 3 y Gráfico 4) muestran la evolución de los atributos de los grafos de amigos y de seguidores durante el periodo de extracción:

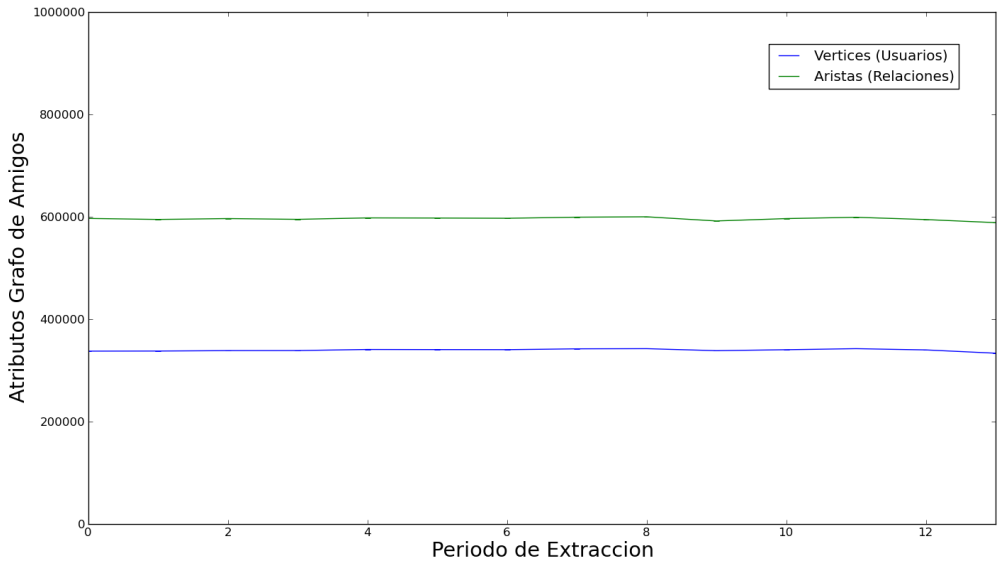


Gráfico 3: Evolución de los atributos de los grafos de amigos

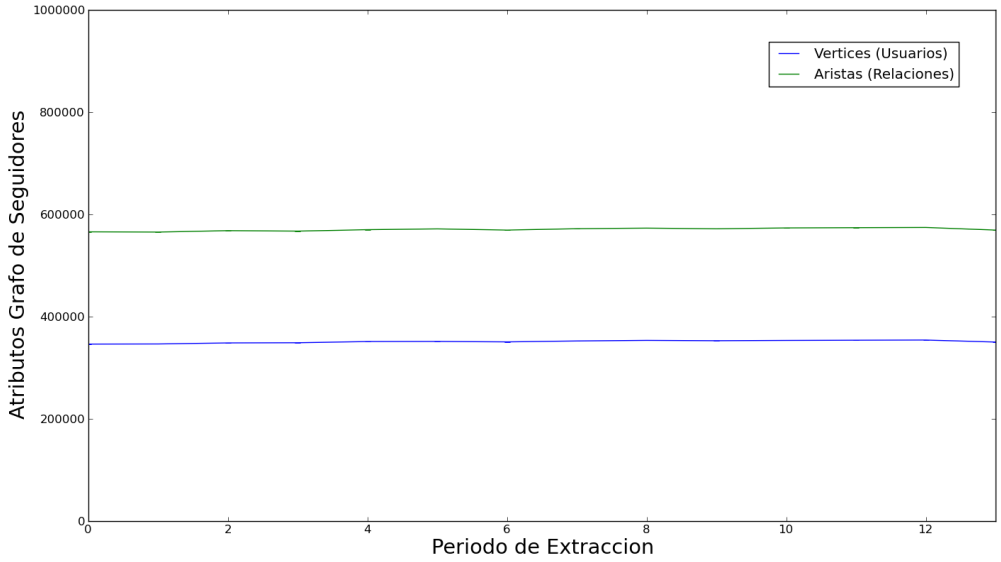


Gráfico 4: Evolución de los atributos de los grafos de seguidores

Según los datos recogidos reflejados en los gráficos anteriores, los grafos de amigos y de seguidores se componen inicialmente de 337.640 y 346.266 vértices o

usuarios, y de 596.794 y 565.859 aristas o relaciones respectivamente. Como puede apreciarse, la composición interna de los grafos no sufre grandes variaciones en cuanto al número de vértices y aristas, con crecimientos y decrecimientos medios que no superan el 0,1% en valor absoluto. Tras los catorce periodos de extracción se obtienen unos promedios de variación del número de usuarios del -0,09% para el grafo de amigos y del 0,09% para el grafo de seguidores, lo que se traduce en un conjunto de información estable y óptimo para el estudio. Por su parte, la variación del número de aristas es del -0,10% y del 0,05% respectivamente. Tanto el porcentaje de variación de número de usuarios como el de número de aristas se deben a alteraciones en la estructura de la red social, así como a pequeños márgenes de error aceptables producidos durante el proceso de extracción.

6.2.2. Análisis del Grado Nodal

El grado nodal o grado de un vértice se trata de la primera y más simple definición de centralidad en la teoría de grafos. Se define como el número de aristas que inciden en un vértice. En el contexto de una red social equivale al número de relaciones que posee un usuario determinado, y en el caso particular del proyecto el número de amigos o de seguidores.

En el análisis de grado nodal de los grafos sociales extraídos de *Twitter*, se ha calculado la distribución de las relaciones de amistad o de seguimiento entre todos los nodos o usuarios. Es decir, el número de relaciones internas que posee cada usuario y la posibilidad de clasificarlos en varios grupos en función del tamaño de dicho número de relaciones. Más concretamente se ha estudiado el porcentaje de usuarios que poseen un mayor número de amigos o de seguidores dentro de un grupo determinado, así como los que poseen un número menor, y la relación entre ambos colectivos.

Para ello se han calculado las distribuciones de valores mínima, media y máxima durante el calendario de extracción, tanto para el grafo de amigos como para el

grafo de seguidores. Dichas series equivalen al número mínimo, medio y máximo de usuarios respectivamente para todos los periodos, que posean de 0 a n número de relaciones de amistad-seguimiento, con intervalos de una unidad. Es decir, el número de usuarios con 0, 1, 2, 3, etc. relaciones hasta llegar al número máximo de relaciones internas por usuario. Para poder visualizar detalladamente los resultados obtenidos, se han representado en un gráfica con ejes logarítmicos: x para el número de relaciones de amistad o de seguimiento en función del grafo del que se trate, e y para el número de usuarios. Los valores del eje x para el número de relaciones parten desde el valor 0 hasta 1.000, mientras que los del eje y para el número de usuarios desde 0 hasta 1.000.000.

Las distribuciones de valores mínima, media y máxima para el grafo de amigos durante el periodo de extracción pueden observarse a continuación:

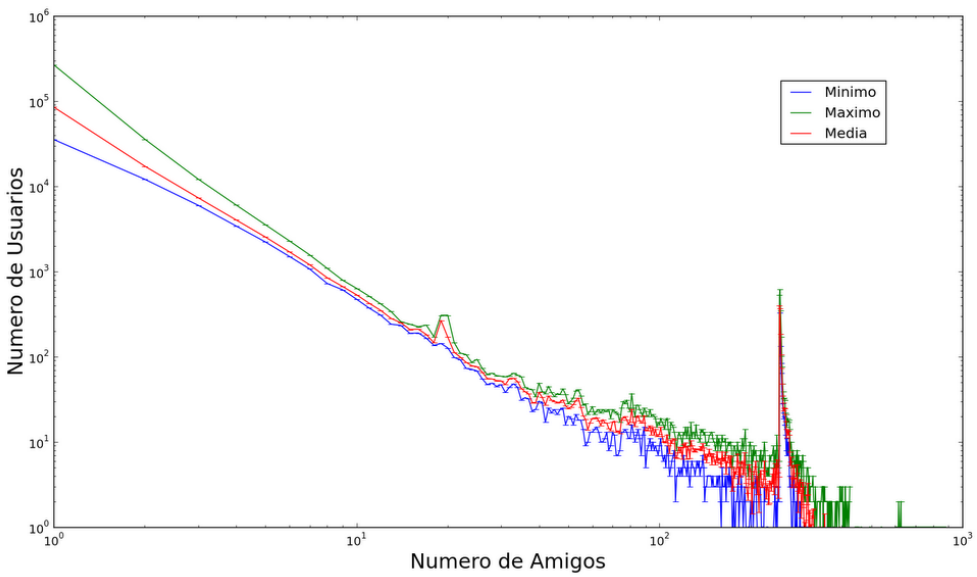


Gráfico 5: Grado nodal del grafo de amigos

Por su parte, las series mínima, media y máxima para el grafo de seguidores durante el periodo de extracción pueden visualizarse en el Gráfico 6.

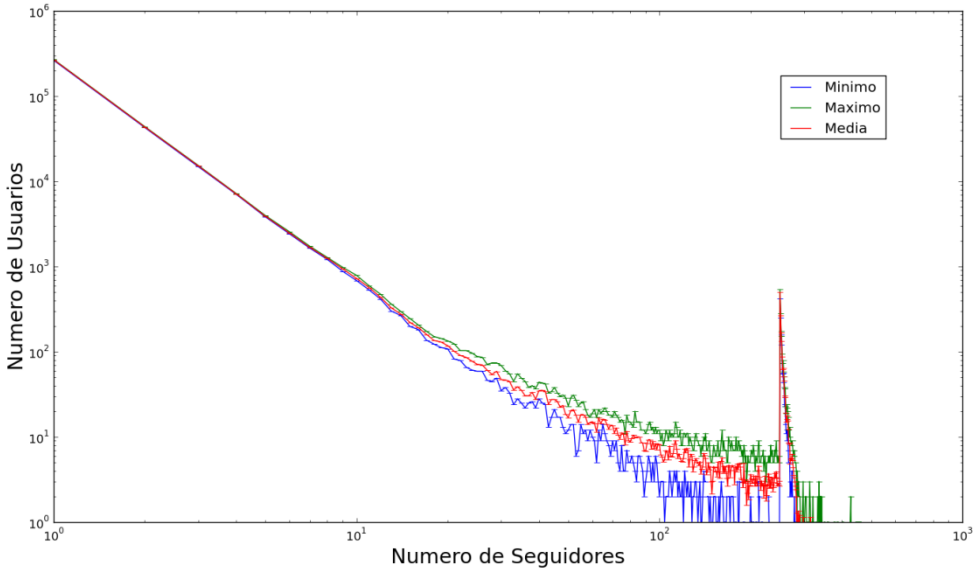


Gráfico 6: Grado nodal del grafo de seguidores

Como puede contemplarse ambas gráficas presentan un aspecto similar, que muestra una distribución particular de las relaciones de amistad o de seguimiento entre todos los usuarios. Claramente se observa como un porcentaje muy elevado de usuarios (en torno al 80%) posee un número muy reducido de relaciones, mientras que el pequeño porcentaje restante (alrededor del 20%) acapara la gran mayoría. Este tipo de distribución forma parte de las llamadas distribuciones "power law"[70], basadas en el *Principio de Pareto,* aplicable a múltiples y diversas situaciones en distintos campos.

Vilfredo Pareto[71] fue un sociólogo, economista y filósofo italiano nacido en el siglo XIX, famoso por su observación de que en Italia el 20% de la población poseía el 80% de los recursos, reflexión que posteriormente *Joseph Juran*[72] y otros popularizarían con el nombre de *Principio de Pareto*[73], dando lugar al concepto de *Distribución de Pareto*[74]. Este principio es también conocido como la regla del 80-20, y está basado en el conocimiento empírico, es decir, resulta de la observación y de la experiencia sin una base científica.

La siguiente gráfica es una representación del *Diagrama de Pareto* en el contexto en el que fue concebido, las clases sociales italianas de comienzos del siglo XIX:

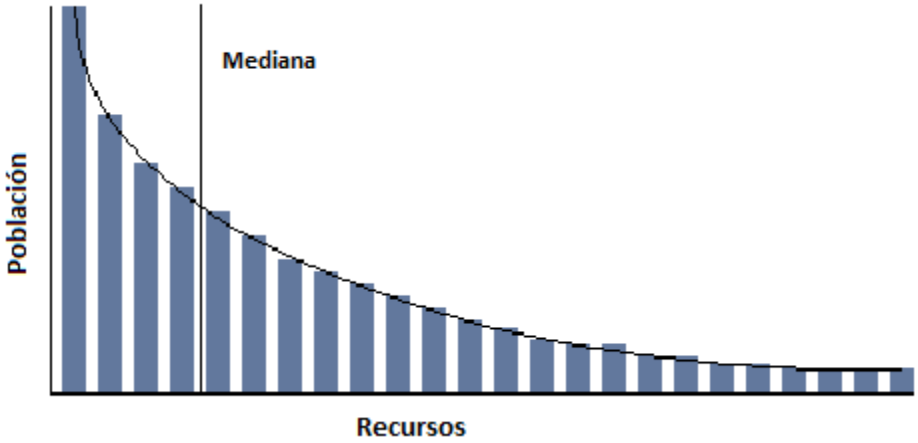


Gráfico 7: Diagrama de Pareto

Los resultados obtenidos del análisis del grado nodal, demuestran que el *Principio de Pareto* es ciertamente aplicable en los grafos de amigos y seguidores de *Twitter* extraídos para el proyecto (otros estudios[75][76][77] han señalado la existencia de distribuciones "power law" en diversas comunidades sociales). Haciendo un paralelismo con el ejemplo que *Pareto* tomo en su día, las relaciones de cada usuario pueden considerarse los recursos para los que *Pareto* observo una distribución desigual en la sociedad italiana de su época. Más adelante, en el subapartado 6.2.4 se muestra como el número de relaciones de un nodo es determinantes a la hora de conseguir otras nuevas, lo que explica en gran parte el tipo de distribución encontrada. Los usuarios con un gran número de relaciones tienen una mayor facilidad para conseguir otras nuevas, y de forma inversa con los usuarios con un número menor.

Finalmente, en cuanto al pico observado en el número de usuarios (eje y) que presentan ambas gráficas en torno al valor del 250 para el número de relaciones (eje x), éste se debe a la restricción de 250 relaciones por usuario para la muestra de extracción inicial, ya señalada en el subapartado 0, y por lo tanto no debe tenerse en cuenta. Es una consecuencia de los parámetros del proceso de extracción, y se explica como la agrupación todos los usuarios con más de 250 relaciones fuera de la muestra

de estudio. Por otra parte, como ya se ha mencionado con anterioridad, el hecho de que en las gráficas puedan observarse valores superiores a 250 para el número de relaciones se debe a la aparición de nuevas relaciones internas durante el calendario de extracción.

6.2.3. Análisis del Clustering Coefficient

El c*lustering coefficient* o coeficiente de agrupamiento de un vértice en un grafo es una medida de agrupamiento concebida por *Duncan J. Watts*[51]. Equivale al número de conexiones existentes entre los nodos vecinos de un vértice, dividido entre el número total de conexiones que podrían existir entre ellos, y por lo tanto su valor mínimo es cero y máximo uno. De forma genérica, cuando se menciona el término c*lustering coefficient* se hace referencia al *local clustering coefficient* para un vértice determinado, mientras que el *network average clustering coefficient* equivale a la media de los *local clustering coefficients* de todos los vértices del grafo.

Para una mejor comprensión del concepto de c*lustering coefficient* y mostrar algunos ejemplos de su cálculo se presenta la siguiente figura.

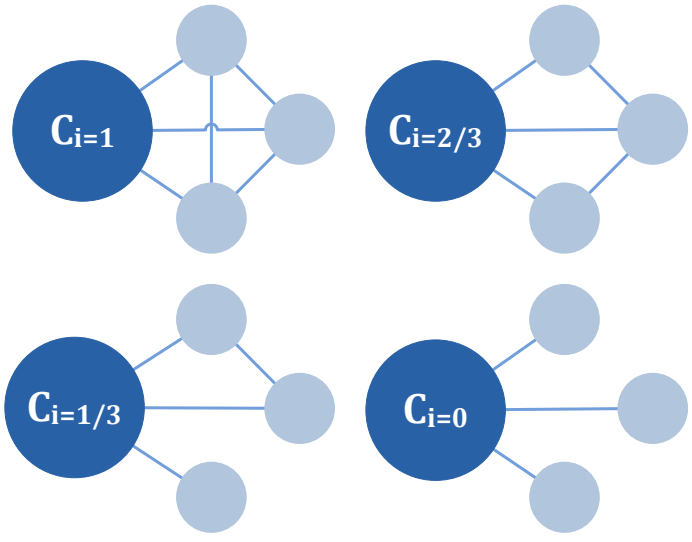


Figura 20: Ejemplo de cálculo del *clustering coefficient*

En ella se contemplan los cuatro posibles valores de c*lustering coefficient* para un nodo conectado con otros tres vértices vecinos, variando el número de conexiones

existentes entre ellos. Partiendo desde las máximas posibles hasta eliminar todas ellas, se observa como el valor decrece de 1 a 0 pasando por 2/3 y 1/3.

En el contexto del proyecto, la importancia del cálculo del c*lustering coefficient* para los grafos de amigos y de seguidores extraídos del servicio de red social *Twitter*, radica en conocer su grado de interconectividad general, o lo que es lo mismo, el nivel de desarrollo de las relaciones internas entre los usuarios. En forma de ejemplo práctico, elevados valores de c*lustering coefficient* podrían significar que se trata de un grafo en el que un alto porcentaje de usuarios comparten relaciones de segundo grado.

Este mecanismo de establecimiento de nuevas relaciones entre usuarios es muy común en muchos servicios de redes sociales, ya que éstos prestan herramientas específicas con dicho propósito. En el servicio de red *Facebook* por ejemplo, existe un sugeridor[78] de amigos que proporciona al usuario una lista de otros usuarios que están relacionados concurrentemente con sus amistades actuales, o comparten aficiones e intereses. De esta forma, se consiguen enriquecer las relaciones internas entre nodos dentro de un conjunto determinado de usuarios, y por consiguiente aumentar sus valores de c*lustering coefficient*.

El servicio de red social *Twitter* no dispone de ninguna herramienta similar que agilice el proceso de búsqueda de amigos potenciales, en todo caso el usuario puede consultar los amigos y seguidores de sus conexiones para encontrar convergencias. Éste hecho, entre otros, es uno de los factores que explica los resultados obtenidos durante el estudio de los grafos de amigos y seguidores que se exponen a continuación.

Con objeto del análisis, se han hallado los c*lustering coefficient* de cada uno de los vértices de ambos grafos por separado, y los resultados obtenidos han arrojado unos muy altos porcentajes de usuarios con valores de *clustering coefficient* igual a

cero. En concreto, para el grafo de amigos el porcentaje ha sido del 99,65% mientras que para el grafo de seguidores del 99,85%.

Por otra parte, el reducido número de usuarios con *clustering coefficient* de valor uno está compuesto principalmente por una serie de nodos que cumplen alguna de las siguientes condiciones:

- **Forman parte del conjunto raíz** y poseen **un pequeño número de relaciones**. En ese caso, es muy probable que los nodos relacionados pertenezcan de igual modo al conjunto raíz y sus relaciones también hayan sido incluidas en la muestra del estudio.
- **Representen un contacto común** entre un pequeño número de usuarios, que a su vez se encuentren interconectados. Por ejemplo, el usuario A es seguido por todos los miembros del grupo compuesto por B, C y D, que a su vez están conectados. Si bien este es un hecho que no ha podido ser comprobado a partir de los datos extraídos, es muy posible que las relaciones con A dentro del grupo se hayan propagado a partir del primero de sus miembros que se relacionó con A. En términos sociales, equivaldría al conjunto de personas que pasaron a relacionarse con todos los miembros de un grupo ya establecido.

Dados los resultados obtenidos, para una mejor apreciación de los valores de *clustering coefficient* distintos de cero, éstos se han representado mediante gráficas de tipo CDF (Cumulative Distribution Function). Dichas gráficas presentan un eje x para los valor*es de clustering coefficient* contenidos en el intervalo (0,1], y un eje y con el porcentaje acumulado de usuarios.

Las gráficas CDF (Gráfico 8 y Gráfico 9) para el c*lustering coefficient* de los grafos de amigos y de seguidores pueden observarse a continuación:

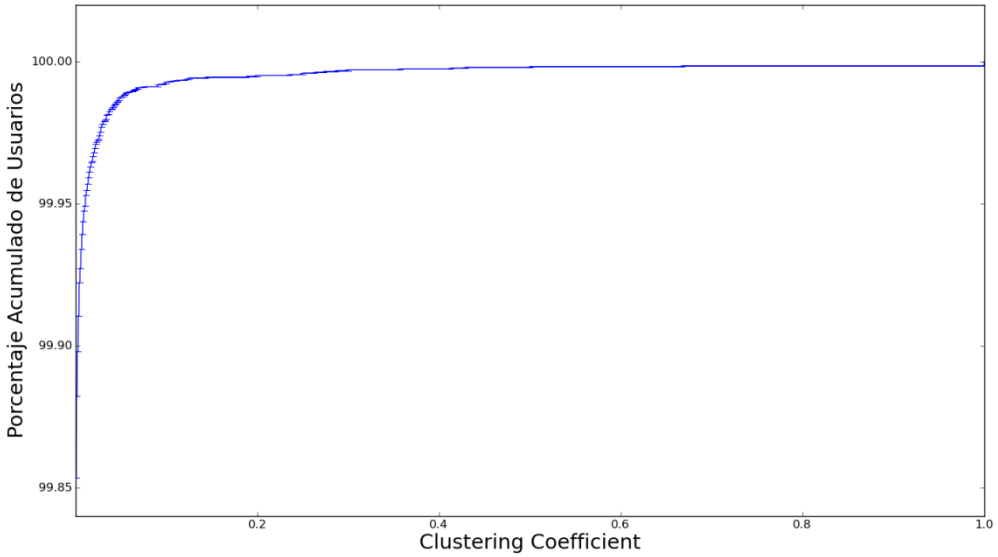


Gráfico 8: CDF para el clustering coefficient del grafo de amigos

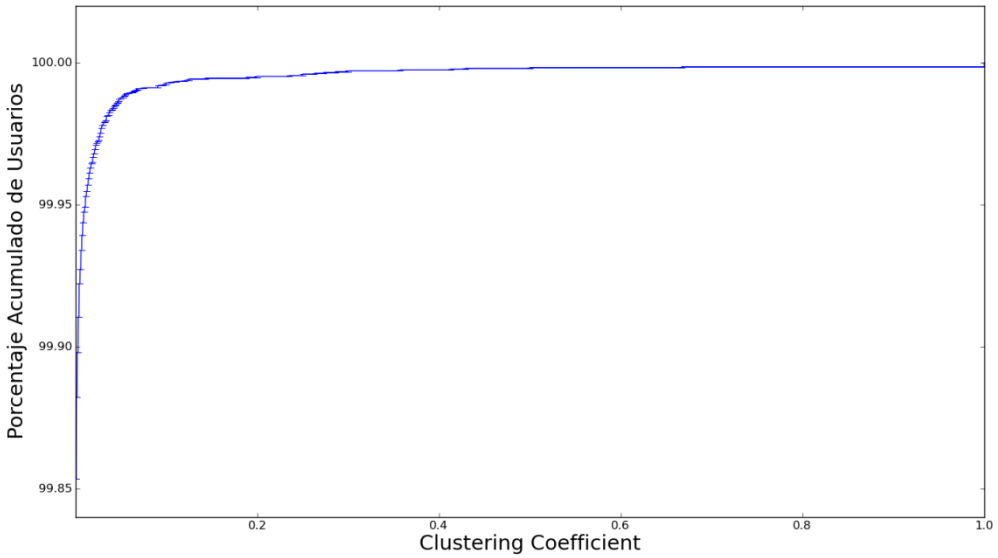


Gráfico 9: CDF para el clustering coefficient del grafo de seguidores

Como puede comprobarse en las anteriores gráficas, los valores de c*lustering coefficient* son muy bajos para la gran mayoría de los nodos que componen el grafo de amigos y el grafo de seguidores, si bien son mínimamente superiores en el primero que en el segundo. Debe tenerse en cuenta que estos datos se encuentran, en cierta

medida, condicionados por la limitación de 250 como el máximo número de amigos o seguidores, ya mencionada en el subapartado 6.1. Ello ha supuesto que no hayan podido ser recuperadas el total de las ramificaciones de los grafos extraídos del servicio de red social *Twitter*. Sin embargo, los resultados son lo suficientemente manifiestos como para arrojar una conclusión clara: el grado de agrupamiento nodal de los grafos extraídos de *Twitter* es muy pequeño. Por otra parte, tales resultados van en la misma dirección de otros alcanzados en estudios[79][80] similares acerca de *Twitter*, y confirman la validez de la conclusión.

6.2.4. Análisis Evolutivo

Uno de los puntos más interesantes en el análisis de los grafos sociales extraídos, es el estudio de la evolución del comportamiento de sus usuarios durante un periodo de tiempo determinado, en este caso durante el calendario de extracción (13/12/2009 – 26/12/2009). Es por ello por lo que se han realizado extracciones periódicas de información con la intención de captar el comportamiento dinámico de los usuarios, y no reducir el estudio a una observación meramente estática. En concreto, mediante el análisis evolutivo de los grafos se ha tratado de averiguar qué cambios se producen en el número de relaciones de amistad o de seguimiento de cada uno de los nodos que los componen, y comprobar si puede extraerse alguna conclusión al respecto.

De acuerdo con los resultados obtenidos en el análisis de grado nodal de los grafos extraídos (ver subapartado 6.2.2), la fuerte dicotomía en la clasificación de los usuarios induce a pensar que la facilidad de ciertos usuarios para conseguir nuevas relaciones depende de su número de relaciones actual. Es decir, aquellos usuarios que parten con una cifra elevada de amigos o seguidores, son más propensos que el resto a aumentar dicha cifra en un futuro próximo. Ello explicaría la existencia de dos grupos de usuarios con características tan marcadas: unos con muchos amigos-seguidores y otros con muy pocos. Sin embargo, esta hipótesis requiere una demostración basada

en los datos reales para lo cual se han generado una serie de gráficas de nube de puntos que contrastan ambos valores.

Tanto para el grafo de amigos como para el grafo de seguidores, se ha evaluado el aumento de relaciones por número actual de relaciones y la disminución de relaciones por número actual de relaciones. Todas las gráficas se han calculado contrastando los valores obtenidos al inicio (13/12/2009) y final (26/12/2009) calendario de extracción, y para el caso especial del análisis evolutivo de los grafos de amigos y de seguidores, se han tomado no solo las relaciones internas de cada usuario o nodo, sino el total de amigos y seguidores de la red *Twitter*.

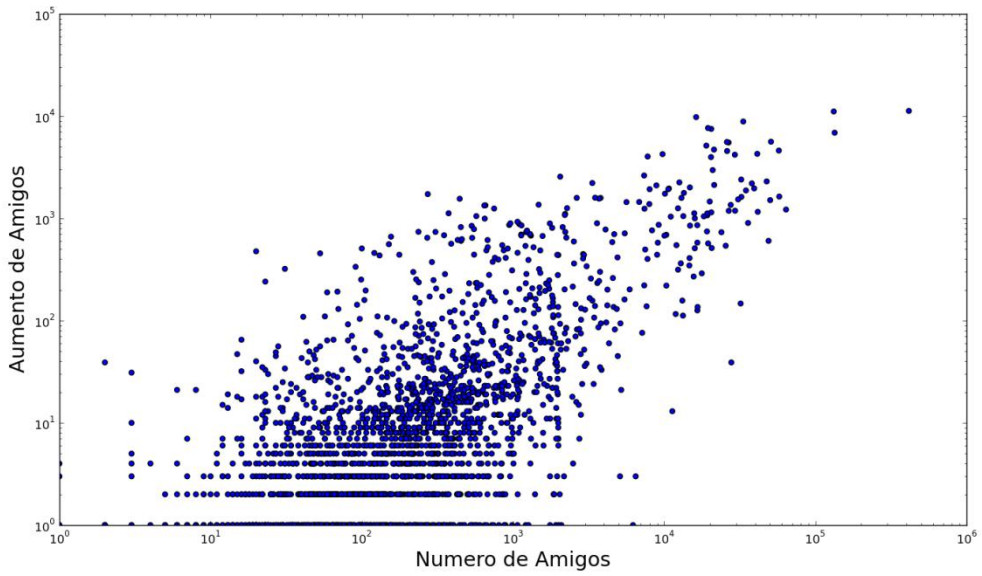


Gráfico 10: Aumento de amigos por número actual de amigos

Como muestra el Gráfico 10, existe una importante relación entre el aumento del número de amigos y el número de amigos actuales. Esta circunstancia es fácilmente explicable, dado que el hecho de que un usuario posea un gran número de amigos actuales implicaría que es un usuario activo de *Twitter,* y por lo tanto todo hace pensar que tienda a aumentar su círculo de amistades en la red. De esta forma, cuanto más activo sea un usuario, mayor será el crecimiento de su número de amigos. Por el contrario, el fenómeno de pérdida de amigos reflejado en el Gráfico 11 es mucho más le-

ve, dado que resulta más extraño que un usuario elimine de su lista de amigos a alguien que decidió incluir con anterioridad.

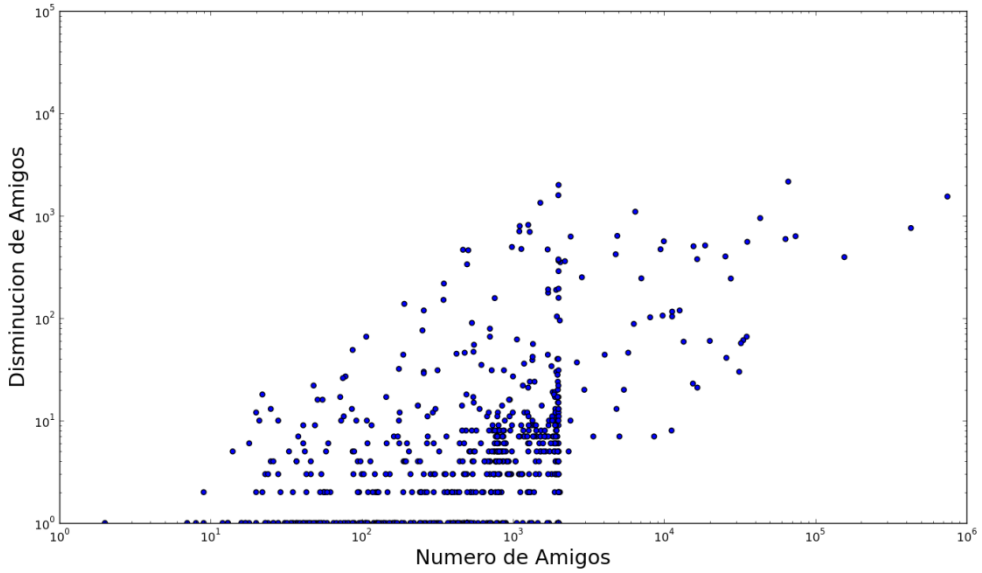


Gráfico 11: Disminución de amigos por número actual de amigos

Por su parte el Gráfico 12 muestra una estrecha relación entre el aumento de seguidores y el número de seguidores actuales, aún mayor que la observada en el grafo de amigos. En este caso, la explicación lógica proviene del grado de exposición al público de *Twitter* que tienen los usuarios cuanto mayor sea su número de seguidores. Un usuario con muchos seguidores, aparecerá recurrentemente en páginas del servicio de red social *Twitter* por diversos motivos: menciones, referencias, vínculos en perfiles, etc. Este hecho permitirá que pueda darse a conocer fácilmente, y por consiguiente un mayor número de usuarios se encuentren predispuestos a pertenecer a su lista de seguidores. En cuanto a la disminución de seguidores por número actual de seguidores del Gráfico 13, los valores son mucho menores. Esto se explica porque la propia naturaleza de los usuarios muy populares o con muchos seguidores, conduce a que estos busquen siempre incrementar el número de sus seguidores y no al contrario.

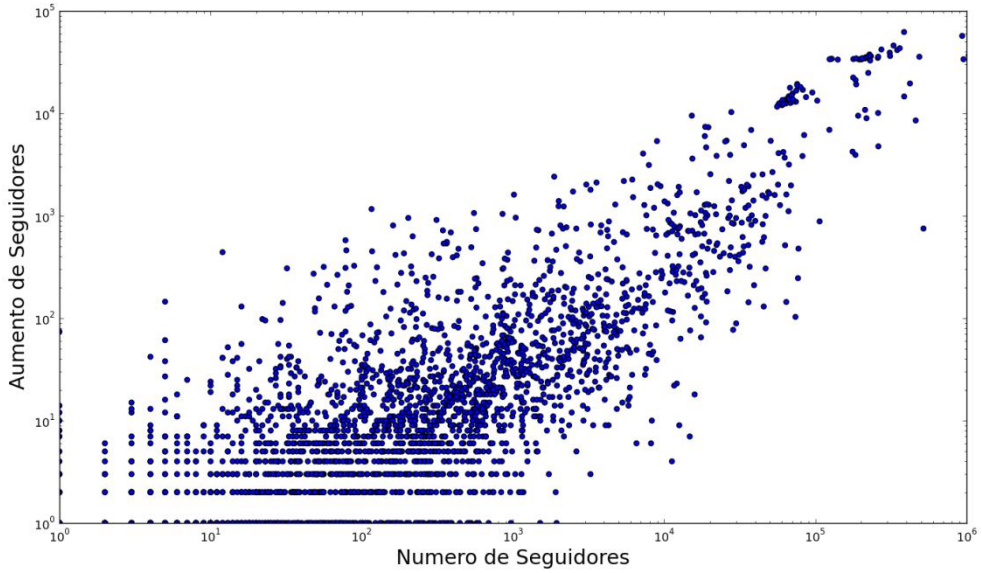


Gráfico 12: Aumento de seguidores por número actual de seguidores

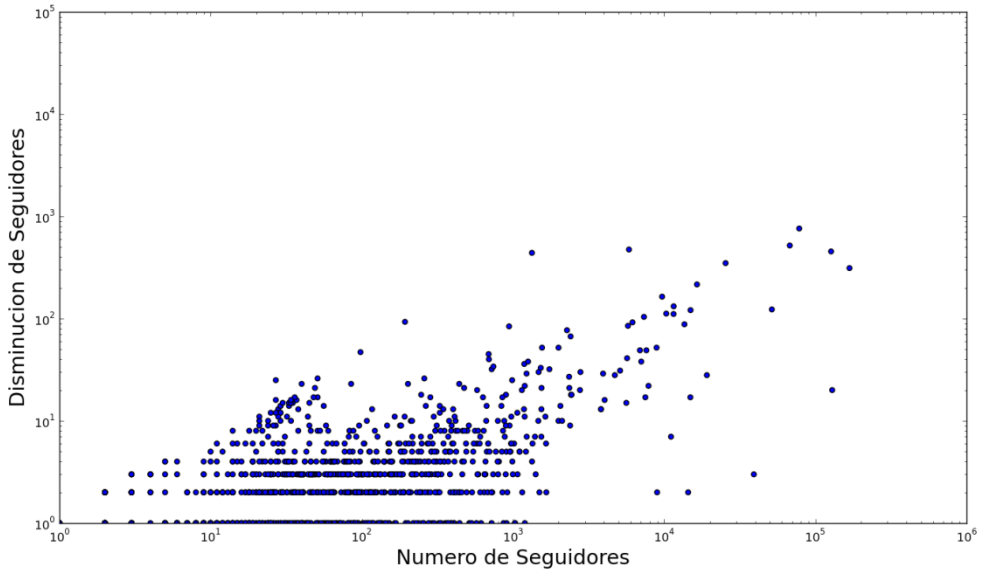


Gráfico 13: Disminución de seguidores por número actual de seguidores

En general, todas las gráficas mostradas verifican que la hipótesis planteada se cumple para los datos extraídos del servicio de red social *Twitter*: la variación en el número de relaciones es fuertemente dependiente del número de relaciones actual, especialmente en el aumento de las mismas. Los usuarios que a día 13/12/2009 ya

disponían una cantidad elevada de amigos o seguidores aumentaron en mayor medida que el resto dichas cantidades hasta el día 26/12/2009. Las gráficas de nubes de puntos, especialmente las de aumento por número actual de relaciones, presentan una disposición de flecha dirigida a la esquina superior derecha de la gráfica. Es decir, los puntos con mayor valor para alguno de los ejes corresponden con también los valores mayores para el otro eje. La relación entre ambos ejes es por tanto directamente proporcional: al aumentar el valor de un eje, aumenta al mismo tiempo el valor del otro.

La siguiente figura ilustra la relación entre los valores de ambos ejes, y como es la disposición general de la nube de puntos para la representación del aumento de relaciones por número de relaciones actuales.

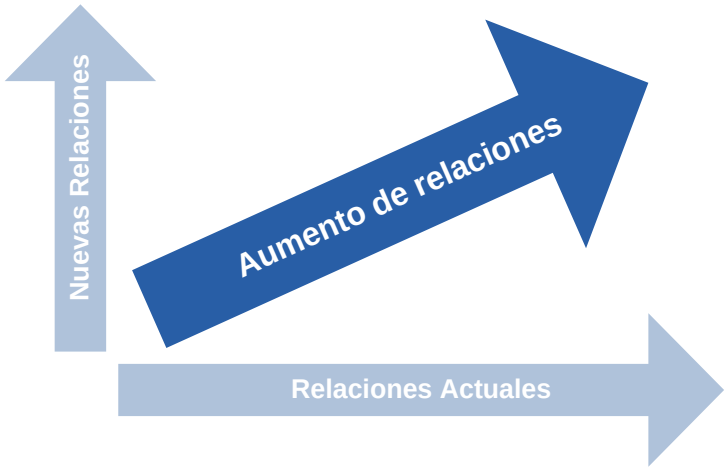


Figura 21: Relación entre nuevas relaciones y relaciones actuales

Por otra parte, este hecho es fácilmente explicable en el contexto de la red social *Twitter* teniendo en cuenta su interfaz gráfica y la forma en la que se presentan los perfiles de los usuarios. En todo perfil se muestran los amigos y seguidores de su dueño, y por lo tanto un usuario con muchos amigos y seguidores aparecerá en el mismo número de perfiles con un enlace a su propio perfil. De esta forma dicho usuario estará más expuesto que otros al público general de *Twitter*, o lo que es lo mismo otros usuarios podrán ser conscientes de su existencia con mayor facilidad y por consiguiente podrán optar a establecer relaciones con él con cierta predisposición. Empleando un

símil con el mundo de la comunicación, generalmente cuanto más publicitado es un producto es más consumido.

La demostración de la hipótesis viene a señalar que dentro de *Twitter* también se cumple un principio utilizado en el contexto económico de la distribución de la riqueza conocido como "the rich get richer". Es decir, los ricos se hacen más ricos o tienen mayor predisposición a aumentar su riqueza en mayor medida que aquellas personas de posición económica inferior. Este principio ha sido utilizado en diversos momentos de la historia, épocas y lugares, y puede decirse que se ha demostrado empíricamente en cada uno de ellos. Ciertos estudios sostienen que ello no es más que una manifestación de la importancia de las conexiones.

6.3. Análisis del Seguimiento de Usuarios

El seguimiento de usuarios tiene por objetivo analizar los "tweets" de dos usuarios particulares durante el calendario de extracción, así como los "tweets" de sus seguidores para analizar la posible influencia sobre estos. Como ya se ha descrito anteriormente, se ha optado por tomar un usuario con un número de amigos y seguidores relativamente pequeño (*zaquito*[67]), y otro mayor (*googlappengine*[68]), para así poder estudiar la influencia en *Twitter* de dos tipos de usuarios diferentes.

Con objeto de medir dicha influencia se ha analizado el número de "retweets", respuestas y menciones de los que han sido objeto los usuarios del estudio por parte de sus seguidores, así como la presencia de "topics" en los "tweets" de los seguidores de *zaquito* y *googlappengine*, que éstos últimos hayan tratado con anterioridad y por lo tanto se pueda suponer cierto grado de influencia (en lo que sigue, este fenómeno concreto se referirá como "topic" propagado). Los resultados obtenidos se presentan en cuatro gráficas comparativas donde se contraponen los números de "retweets", respuestas, menciones y los "topics" propagados de ambos usuarios a lo largo de todo el calendario de extracción, desde el primer periodo (13/12/2009) hasta el vigesimoc-

tavo (09/01/2010). Adicionalmente con apunte común a las gráficas mostradas, ha de destacarse su aspecto periódico, especialmente del Gráfico 14, el Gráfico 15 y del Gráfico 16. Este hecho puede explicarse lógicamente por la mayor actividad de los usuarios del servicio de red social *Twitter* durante los fines de semana y menor durante el resto de la semana.

6.3.1. Seguimiento de Retweets

Realizar un "retweet" consiste en reproducir íntegramente el mensaje de otro usuario, incluyendo su nombre. En el caso del Gráfico 14, se observa como *googlappengine* ha sido objeto de un visiblemente mayor número de "retweets" que *zaquito* durante el calendario de extracción.

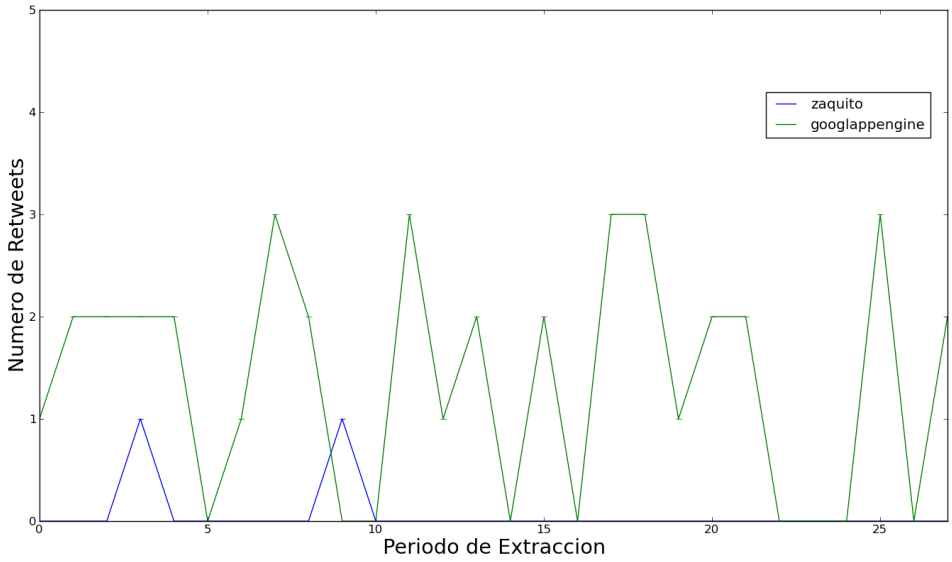


Gráfico 14: Seguimiento de "retweets" de zaquito y googlappengine

Éste fenómeno es fácilmente explicable dada la naturaleza del usuario *googlappengine*: se trata de un canal de noticias y comentarios relacionados con *GAE*. El hecho de que mediante *googlappengine* se publiquen noticias o hechos que puedan tener cierto interés público, provoca que un número de usuarios trate de propagar

dichos "tweets" mediante "retweets" en su perfil. Por lo tanto en este caso, la diferencia en el número de "retweets" entre *zaquito* y *googlappengine* está claramente determinada por el tipo de publicaciones del segundo, unido a su mayor grado de popularidad o mayor número de seguidores.

6.3.2. Seguimiento de Respuestas

Como respuesta dentro del servicio de red social *Twitter*, se entiende la réplica al "tweet" de otro usuario de forma pública. Por lo tanto implica cierto grado de intercambio de comunicación entre usuarios, y es lo más parecido a una conversación clásica en *Twitter*.

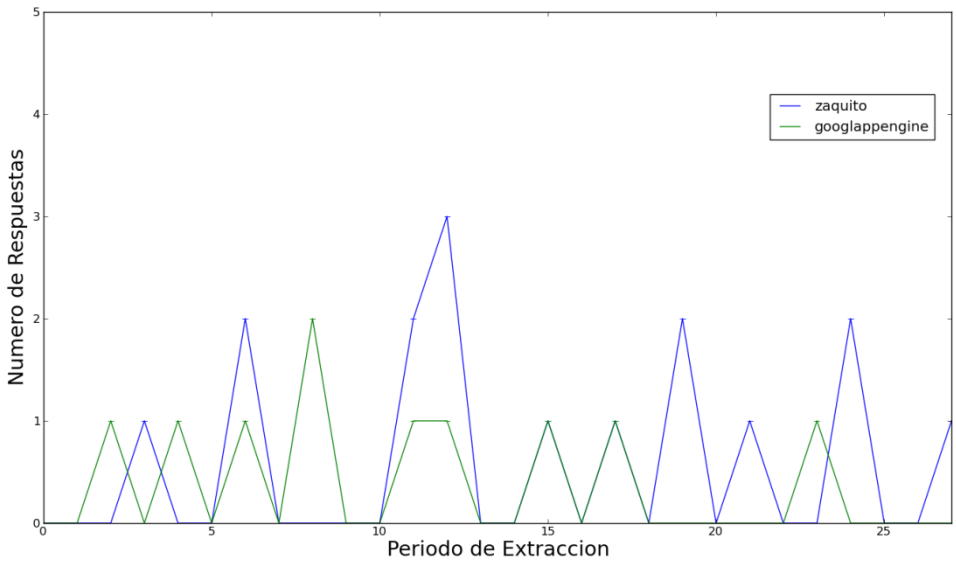


Gráfico 15: Seguimiento de respuestas de zaquito y googlappengine

En este caso los valores obtenidos para los dos tipos de usuarios son mucho más parecidos, siendo incluso mayor el número de respuestas recibidas por *zaquito* que por *googlappengine*. Tal y como ocurre con el número de "retweets", la naturaleza del usuario es determinante en el número de respuestas, pero en esta ocasión a la inversa. A pesar de ser un usuario más popular que *zaquito*, *googlappengine* posee un

perfil menos personal que el primero ya que no representa a ningún individuo concreto. Por su parte, el usuario *zaquito* constituye lo contrario: el perfil de una persona que está en contacto con sus amistades directas e interactúa con ellas mediante "tweets" y respuestas de forma frecuente. Por todo ello no resulta extraño que aún siendo menos popular, *zaquito* reciba más respuestas que *googlappengine*.

6.3.3. Seguimiento de Menciones

En el contexto de Twitter una mención equivale a incluir en un "tweet" el nombre de un usuario precedido por el signo @.

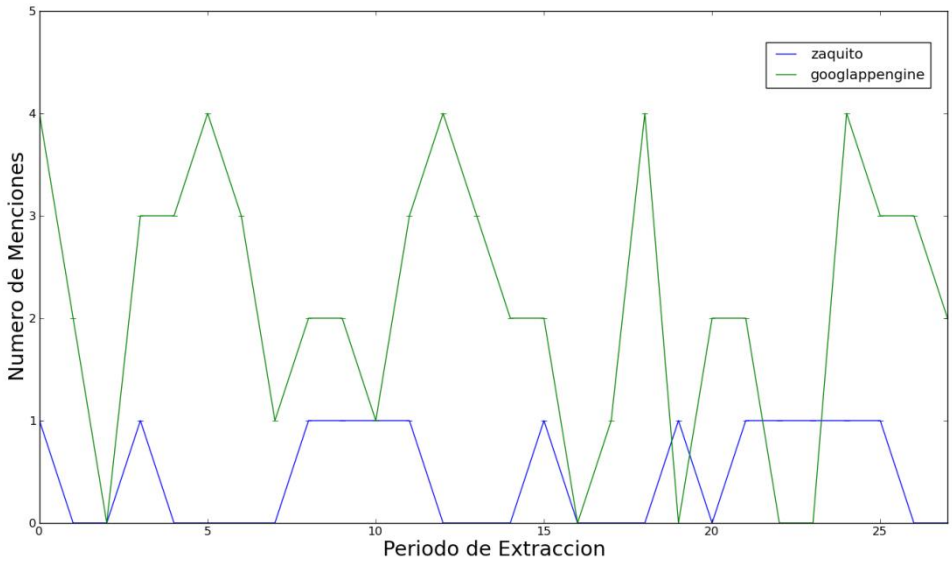


Gráfico 16: Seguimiento de menciones de zaquito y googlappengine

De forma lógica cabe esperar que los usuarios más referenciados o mencionados sean aquellos usuarios más populares, como puede ocurrir en cualquier otro contexto. Twitter no es una excepción y ello explica que el número de menciones recibidas por googlappengine durante el periodo de extracción, sea mucho mayor que el del usuario menos popular *zaquito*. Tal y como se observa en el Gráfico 16, el número de menciones y el grado de popularidad tienen una relación directamente proporcional:

cuanto mayor sea el número de seguidores de un usuario, mayor será su número de menciones.

6.3.4. Seguimiento de Topics Propagados

El caso de los ya descritos "topics" propagados, es algo diferente a los anteriores. A pesar de que solo se han contabilizado "topics" previamente mencionados por los usuarios *zaquito* y *googlappengine*, no puede sentenciarse que sus seguidores hayan sido influenciados directamente por ellos para incluirlos en sus "tweets".

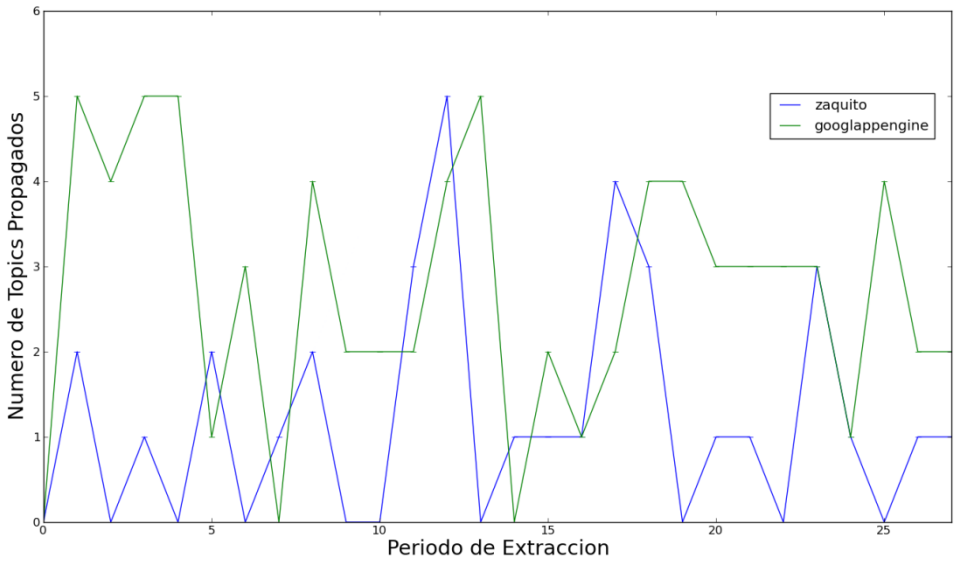


Gráfico 17: Seguimiento de "topics" propagados de zaquito y googlappengine

Tanto para *zaquito* como para *googlappengine* se han recogido relativamente altos valores de topics propagados como puede observarse en el Gráfico 17. Sin embargo, la gran mayoría de estos "topics" propagados hacen referencia a términos genéricos como por ejemplo *#xmas*, *#navidad*, *#newyear* o *#2010*, de los que resulta imposible averiguar el verdadero origen. Por lo tanto, en esta sección concreta del estudio no pueden enunciarse conclusiones ya que los resultados obtenidos no lo permiten.

7. Planificación

En el siguiente apartado se aborda la planificación del proyecto atendiendo a su calendario de ejecución y al presupuesto de los recursos empleados.

7.1. Calendario

Además del calendario completo de ejecución del proyecto, se ha incluido un detalle del calendario de extracción y seguimiento, en el que se desglosan los periodos de extracción de los grafos sociales y de seguimiento de los usuarios del servicio de red social *Twitter.*

7.1.1. Calendario de Ejecución

La ejecución del proyecto *Extracción y análisis de información del servicio de red social Twitter, a través de la plataforma de "cloud computing" Google App Engine* ha tenido una duración de 23 meses desde Junio de 2009 hasta Mayo de 2011. Todo el proyecto puede dividirse en una serie de fases y subfases que se han ejecutado de acuerdo a un calendario y atendiendo ciertas dependencias entre ellas.

Las principales fases y subfases que componen el calendario de ejecución del proyecto son las siguientes:

Fase	Fase preliminar
Dependencias	-
Duración	3 meses (Junio 2009 – Septiembre 2009)
Subfases	• **Elección de la temática y tutores** a partir los temas disponibles en el tablón de proyectos de fin de carrera de la *Escuela Politécnica Superior de la Universidad Carlos III de Madrid*. • **Estudio del estado del arte** acerca de los campos del cono-

	cimiento tratados en el proyecto: las redes sociales, el paradigma "cloud computing" y la teoría de grafos. • **Elección del servicio de red social** del que llevar a cabo las extracciones de información: *Twitter*. • **Elección de la plataforma "cloud computing"** para realizar las extracciones de información: *Google App Engine* (*GAE*).

Tabla 17: Fase preliminar

Fase	Planificación y diseño
Dependencias	Fase preliminar
Duración	2 meses (Septiembre 2009 – Noviembre 2009)
Subfases	• **Planificación del proyecto**, incluyendo el calendario de ejecución y una estimación del presupuesto. • **Diseño de los componentes** del entorno "cloud" y del entorno local.

Tabla 18: Planificación y diseño

Fase	Desarrollo y pruebas
Dependencias	Planificación y diseño
Duración	2 meses aprox. (Noviembre 2009 – finales Diciembre 2009)
Subfases	• **Desarrollo de los componentes** del entorno "cloud" y del entorno local. • **Pruebas de los componentes** del entorno "cloud" y del entorno local.

Tabla 19: Desarrollo y Pruebas

Fase	Extracción de la información
Dependencias	Desarrollo y pruebas
Duración	1 mes aprox. (Enero 2010)
Subfases	• **Extracción de los grafos sociales** del servicio de red social *Twitter* mediante los componentes del sistema diseñados para tal efecto y siguiendo el calendario de extracción.

	• **Seguimiento de usuarios** seleccionados del servicio de red social *Twitter* mediante los componentes del sistema diseñados para tal efecto y siguiendo el calendario de seguimiento.

Tabla 20: Extracción de la Información

Fase	Análisis e interpretación de la información
Dependencias	Extracción de la información
Duración	6 meses (Febrero 2010 – Agosto 2010)
Subfases	• **Análisis e interpretación de los grafos sociales** mediante las herramientas *Graph-Tool* y *PyLab*. • **Análisis e interpretación del seguimiento de usuarios** mediante las herramientas propias desarrolladas en *Java* y *PyLab*.

Tabla 21: Análisis e Interpretación de la Información

Fase	Presentación de los resultados
Dependencias	-
Duración	12 meses (Mayo 2010 – Mayo 2011)
Subfases	• **Elaboración de la memoria** de acuerdo a las recomendaciones de los tutores del proyecto y a la guía de elaboración de la *Escuela Politécnica Superior de la Universidad Carlos III de Madrid*. • **Elaboración de la presentación final** para la defensa del proyecto de fin de carrera ante el tribunal.

Tabla 22: Presentación de los Resultados

El Gráfico 18 muestra un diagrama de *Gantt* con las distintas fases y subfases del proyecto desglosadas en el calendario de ejecución.

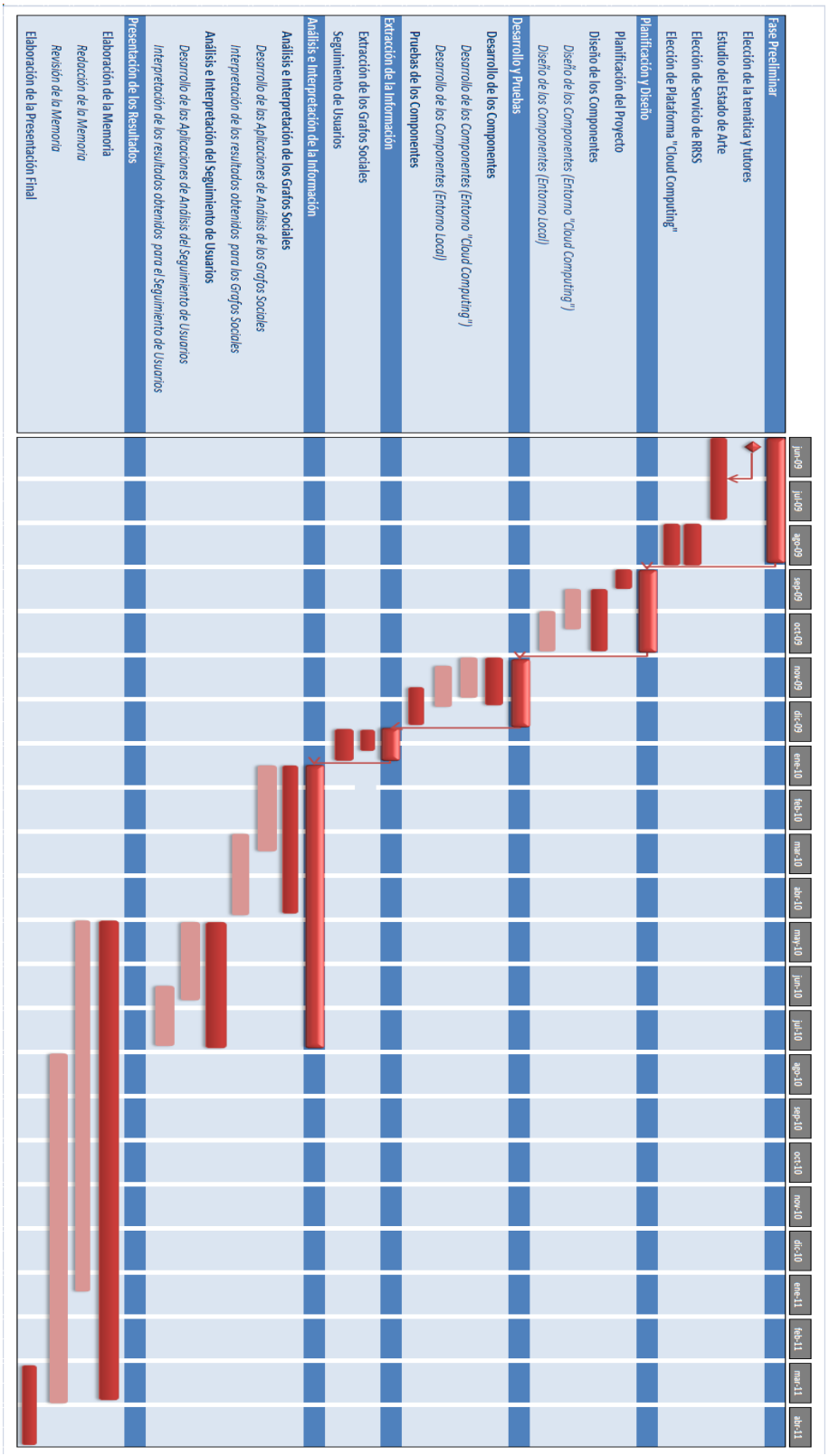


Gráfico 18: Diagrama de Gantt de las fases del proyecto

7.1.2. Detalle del Calendario de Extracción y Seguimiento

La extracción de información de los grafos sociales y el seguimiento de los usuarios del servicio de red social *Twitter*, se ha llevado a cabo según un calendario de extracción y seguimiento. Tanto el proceso de extracción de los grafos sociales, como el del seguimiento de usuarios se iniciaron el día 13/12/2009, si bien el primero tuvo una duración de 2 semanas (14 días) y el segundo de 4 semanas (28 días). En el Gráfico 19 puede apreciarse en detalle el calendario de extracción y seguimiento del proyecto.

Lunes	Martes	Miercoles	Jueves	Viernes	Sábado	Domingo
13/12/2009	14/12/2009	15/12/2009	16/12/2009	17/12/2009	18/12/2009	19/12/2009
Extracción día 1 Seguimiento día 1	Extracción día 2 Seguimiento día 2	Extracción día 3 Seguimiento día 3	Extracción día 4 Seguimiento día 4	Extracción día 5 Seguimiento día 5	Extracción día 6 Seguimiento día 6	Extracción día 7 Seguimiento día 7
20/12/2009	21/12/2009	22/12/2009	23/12/2009	24/12/2009	25/12/2009	26/12/2009
Extracción día 8 Seguimiento día 8	Extracción día 9 Seguimiento día 9	Extracción día 10 Seguimiento día 10	Extracción día 11 Seguimiento día 11	Extracción día 12 Seguimiento día 12	Extracción día 13 Seguimiento día 13	Extracción día 14 Seguimiento día 14
27/12/2009	28/12/2009	29/12/2009	30/12/2009	31/12/2009	01/01/2010	02/01/2010
- Seguimiento día 15	- Seguimiento día 16	- Seguimiento día 17	- Seguimiento día 18	- Seguimiento día 19	- Seguimiento día 20	- Seguimiento día 21
03/01/2010	04/01/2010	05/01/2010	06/01/2010	07/01/2010	08/01/2010	09/01/2010
- Seguimiento día 22	- Seguimiento día 23	- Seguimiento día 24	- Seguimiento día 25	- Seguimiento día 26	- Seguimiento día 27	- Seguimiento día 28

Gráfico 19: Detalle del calendario de extracción y seguimiento

Por otra parte, en el subapartado 12.1 pueden consultarse los recursos *GAE* utilizados durante las fases de extracción del calendario.

7.2. Presupuesto

El presupuesto del proyecto *Extracción y análisis de información del servicio de red social Twitter, a través de la plataforma de "cloud computing" Google App Engine* asciende a la cantidad de, impuestos incluidos, **cuarenta y cuatro mil novecientos ochenta y dos con cuarenta y cuatro (44.982,44 €)**, sin contemplar ningún coste por el uso de los recursos técnicos en el entorno "cloud". En el caso de que dichos recursos no formaran parte de la cuota gratuita de *GAE*, el presupuesto ascendería a la cantidad de cuarenta y cinco mil setenta y tres con uno euros (45.073,01 €).

La Tabla 23 muestra las partidas presupuestarias del proyecto junto con su importe y la Figura 22 una distribución gráfica de las mismas:

Partida	Importe
Recursos personales	44.696,51 €
Recursos técnicos en el entorno local	158,33 €
Recursos técnicos en el entorno "cloud"	90,57 €
Otros costes directos	127,60 €
TOTAL con cuota gratuita de *GAE* (impuestos incluidos)	**44.982,44 €**
TOTAL sin cuota gratuita de *GAE* (impuestos incluidos)	**45.073,01€**

Tabla 23: Presupuesto del proyecto

Figura 22: Distribución del presupuesto del proyecto

Todas las partidas del presupuesto son descritas y detalladas en los siguientes subapartados de la memoria.

7.2.1. Recursos Personales

En la concepción, planificación, ejecución y presentación del proyecto han intervenido, con distinto grado de participación, las siguientes personas:

- ***Guillermo Cabañas Sánchez***: con categoría de Ingeniero, ha participado en condición de autor del proyecto.
- ***Juan Manuel Tirado Martín***: con categoría de Ingeniero Senior, ha participado en condición de tutor del proyecto.
- ***Daniel Higuero Alonso-Mardones***: con categoría de Ingeniero Senior, ha participado en condición de co-director del proyecto.

El número de horas de cada uno de los recursos personales empleados en las distintas fases del proyecto se muestra en la siguiente tabla:

	Guillermo Cabañas	***J. Manuel Tirado***	***Daniel Higuero***
Fase preliminar	200 horas	30 horas	30 horas
Planificación y diseño	300 horas	30 horas	30 horas
Desarrollo y pruebas	150 horas	0 horas	0 horas
Extracción de la información	50 horas	0 horas	0 horas
Análisis e interpretación de la información	300 horas	30 horas	30 horas
Presentación de los resultados	400 horas	50 horas	50 horas
TOTAL	**1.400 horas**	**140 horas**	**140 horas**

Tabla 24: Recursos personales en horas de trabajo

Atendiendo a los totales de horas empleadas por cada uno de los recursos personales del proyecto y a las tarifas[81] para cada categoría establecidas por la *Escuela Politécnica Superior de la Universidad Carlos III de Madrid*, el desglose presupuestario de los recursos personales empleados es el siguiente:

Recurso	**Categoría**	**Dedicación (horas)**	**Coste (€/hora)**	**Coste Total (€)**
Guillermo Cabañas	Ingeniero	1.400	20,52	28.728,00

Juan Manuel Tirado	Ingeniero Senior	140	32,68	4.575,20
Daniel Higuero	Ingeniero Senior	140	32,68	4.575,20
			TOTAL	**37.878,40**
			(+18% IVA)	**44.696,51**

Tabla 25: Recursos personales en euros

Por consiguiente el importe total, impuestos (IVA) incluidos, de los recursos personales empleados en el proyecto asciende a la cantidad de cuarenta y cuatro mil seiscientos noventa y seis cien con cincuenta y uno euros (44.696,51€).

7.2.2. Recursos Técnicos en el Entorno Local

Como concepto de recursos técnicos en el entorno local únicamente se ha empleado un ordenador portátil modelo *Samsung R510* adquirido en el año 2008 por un importe de 1.000€, impuestos (IVA) incluidos. De acuerdo a la fórmula de cálculo de la amortización[81] fijada por la *Escuela Politécnica Superior de la Universidad Carlos III* de Madrid, el desglose presupuestario de los equipos empleados en el entorno local es el siguiente:

Descripción	**Coste (€)**	**% Uso Dedicado al Proyecto**	**Dedicación (meses)**	**Periodos Depreciación**	**Coste Imputable (€)**
Ordenador portátil *Samsung R510*	1.000	50%	19	60	158,33
				TOTAL (IVA incl.)	**158,33**

Tabla 26: Recursos técnicos en el entorno local

En consecuencia el importe total impuestos incluidos de los recursos técnicos en el entorno local empleados en el proyecto asciende a la cantidad de ciento cincuenta y ocho con treinta y tres euros (158,33€).

7.2.3. Recursos Técnicos en el Entorno "Cloud"

A pesar de que durante el proyecto únicamente se han empleado recursos gratuitos de *Google App Engine*, se ha realizado el cálculo de los costes por hacer uso de dichos servicios en el supuesto de que éstos no fueran gratuitos. Para ello se han tomado valores medios de los recursos consumidos durante la fase de extracción de los grafos sociales y la fase de de lanzamiento de las peticiones de recuperación. Estos valores se han registrado a través de la *Consola de Administración* de *GAE* y pueden consultarse en los subapartados 12.1 y 12.2. Por su parte, el coste de los recursos no gratuitos de *GAE* se ha extraído del portal[82] de la plataforma y vienen expresados en dólares.

La Tabla 27 muestra la lista completa de los recursos empleados en el entorno "cloud" junto con su coste asociado, impuestos (VAT) incluidos, en el caso de que no se encontraran incluidos en la cuota gratita de recursos que proporciona *GAE*.

Descripción	Empresa	Coste Imputable
Consumo CPU Extracciones de Grafos Sociales (14 per. X 10 extract. X 4,5 horas = 630 horas)	*Google, Inc.*	630 horas X 0,10 $/hora = 63 $
Ancho de banda entrante Extracciones de Grafos Sociales (14 per. X 10 extract. X 0,01 GB = 1,4 GB)	*Google, Inc.*	1,4 GB X 0,10 $/GB = 0,14 $
Consumo CPU Recuperaciones de Grafos Sociales (14 per. X 10 extract. X 4 horas = 560 horas)	*Google, Inc.*	560 horas X 0,10$/hora = 56$
Ancho de banda entrante Extracciones de Grafos Sociales (14 per. X 10 extract. X 0,01 GB = 1,4 GB)	*Google, Inc.*	1,4 GB X 0,10 $/GB = 0,14 $
Almacenamiento de datos (2 X 10 extract. X 1 GB = 20 GB)	*Google, Inc.*	20 GB X 0,15 $/GB = 3 $
	TOTAL (VAT incl.)	**122,28 $**

Tabla 27: Recursos técnicos en el entorno "cloud"

Por lo tanto, en el caso de que los recursos de *GAE* empleados no se encontrarán incluidos dentro de su cuota gratuita, su coste ascendería a ciento veintidós con veintiocho dólares (122,28 $), equivalentes a noventa con cincuenta y siete euros (90,57 €) para un tipo de cambio de 1 € = 1,35 $.

7.2.4. Otros Costes Directos

Por último, además de los recursos personales y técnicos empleados es necesario incluir el resto de costes directos derivados del proyecto. En este caso estos se reducen a los costes de una conexión ADSL durante los 19 meses de duración del proyecto. Dicha conexión ha sido necesaria para vincular el entorno local y el entorno "cloud", pero también para recopilar información acerca de los campos del conocimiento, plataformas, herramientas y lenguajes relacionados con el proyecto. El uso de la conexión no se ha dedicado exclusivamente al proyecto, por lo que se ha estimado un porcentaje de dedicación del 20%. El desglose presupuestario de los costes directos del proyecto es el siguiente:

Descripción	Empresa	Coste Imputable (€)
Conexión ADSL 10MB (19 meses)	*Telefónica S.A.*	19 meses X 32 €/mes X 20% dedicación = 127,60 €
	TOTAL (IVA incl.)	**127,60**

Tabla 28: Otros costes directos del proyecto

En resumen, el importe total, impuestos (IVA) incluidos, de los costes directos asociados a la ejecución del proyecto asciende a la cantidad de ciento veintisiete con sesenta euros (127,60€).

8. Conclusiones

En el siguiente apartado se abordan tanto las conclusiones generales como las conclusiones personales extraídas durante la realización del presente proyecto.

8.1. Conclusiones Generales

Desde sus inicios, el proyecto que nos ocupa se planteó con dos objetivos claramente diferenciados: realizar la extracción de datos a gran escala a través de una plataforma "cloud computing", y verificar si ciertos principios que se cumplen en las relaciones personales de las sociedades corrientes, tienen su correspondencia en las sociedades paralelas que proliferan a partir de los servicios de redes sociales. Por ello, las conclusiones del proyecto se dividen también en dos bloques.

8.1.1. Uso de Google App Engine

En primer término se ha evaluado la idoneidad del uso de una plataforma "cloud computing", *Google App Engine* en particular, para llevar a cabo extracciones de datos a gran escala. En este sentido, *GAE* se ha mostrado como una buena elección para tales procedimientos por su elevado grado de escalabilidad y elasticidad. Las aplicaciones desarrolladas en *GAE* (extractores de amigos/seguidores) se han comportado satisfactoriamente ante cargas de trabajo cambiantes como son las extracciones de grafos sociales, mientras que su modelo clásico de suministro de servicios relacionaba de manera directa las operaciones realizadas con los recursos consumidos. Por otra parte, el *Balanceador de Carga* de *GAE* ha permitido sortear las restricciones sobre el uso de la API de *Twitter* mediante la asignación dinámica de distintas máquinas para la ejecución de las aplicaciones. Este punto ha resultado clave en todo el proceso, pues sin él no habría podido llevarse a cabo, y de hecho se trata de un mecanismo que podría exportarse a otros procedimientos similares, en los que se requiera utilizar grandes

rangos de direcciones IP. Por ejemplo, la utilidad es evidente en el desarrollo de aplicaciones que consuman de fuentes externas recursos limitados por direcciones IP.

Respecto al resto de funcionalidades ofrecidas por *GAE*, las conclusiones también son positivas destacando principalmente su *Consola de Administración* y su *Sistema de Colas de Tareas*. La *Consola de Administración* ofrece un acceso total a los parámetros de configuración de la aplicación y permite consultar un histórico de las últimas 24 horas, todo ello mediante una interfaz gráfica e intuitiva que ha facilitado enormemente el seguimiento del proceso. El *Sistema de Colas de Tareas*, por su parte, también ha resultado de vital importancia, pues toda la asignación de tareas de extracción de los extractores de amigos y seguidores del entorno "cloud" se articula en torno a él. Cabe señalar asimismo, que la plataforma *GAE* se encuentra en estado de constante desarrollo, por lo que es de esperar que nuevas funcionalidades con una potencial utilidad aparezcan en el futuro a corto plazo. Este hecho, junto con las bondades de la plataforma ya descritas, refuerza su capacidad y conveniencia para determinados usos.

A pesar de que durante el proyecto no se hayan empleado recursos de pago de *Google App Engine*, debido al grado de satisfacción tras su utilización esta opción se presenta como recomendable para proyectos similares en los que pueda disponerse de recursos económicos. El uso de una arquitectura como la que ofrece *GAE*, junto con una cuenta de facturación para los recursos consumidos, resultaría especialmente útil para alojar aplicaciones con necesidades de recursos variables y hasta cierto punto difíciles de prever. Entre las ventajas de *GAE*, y por extensión también de otras plataformas "*cloud computing*" para este tipo de aplicaciones, destacan:

- **La no necesidad de una fuerte inversión inicial** en la infraestructura o servidores en los que alojar las aplicaciones. En la mayoría de los casos este tipo de inversiones exige un cálculo aproximado del número de recursos re-

queridos por la aplicación, y esto es especialmente complicado en aplicaciones cuyos usuarios potenciales sean difícilmente estimables. Para estos casos, resulta difícil valorar la rentabilidad de una fuerte inversión inicial en infraestructura, y por ello el uso de una plataforma "cloud computing" representa una opción atractiva.

- **La capacidad de atender demandas de recursos muy dinámicas** gracias a la escalabilidad de la plataforma. El hecho de que la aplicación pueda ser desplegada en un número variable de servidores y dotada de un diferente volumen de recursos en función de la demanda, permite satisfacer este tipo de necesidades.
- **La relación directa entre los recursos consumidos y los costes** que proporciona la elasticidad de la plataforma. Esta cualidad es especialmente significativa en aplicaciones con propósito comercial en las que los beneficios estén ligados al uso de recursos (número de accesos, peticiones, etc.), y por lo tanto altos niveles de de consumo signifiquen la obtención de pingües beneficios que puedan sufragar los costes. Por el contrario, si no se consumen demasiados recursos los beneficios serán menores, pero también lo serán los gastos por lo que se mantendrá la viabilidad económica de la aplicación.

Como conclusión general acerca del paradigma "cloud computing", y en particular sobre la plataforma *Google App Engine*, puede decirse que nos encontramos ante un concepto que está llamado a tener una importancia creciente en el futuro próximo. Se trata de un paradigma muy adecuado para determinados tipos de aplicaciones, y es de esperar que su número de usuarios sea cada vez mayor. Sin embargo, en el contexto de la tecnología los ciclos de vida suelen ser especialmente cortos, y solo el tiempo será capaz de determinar si nos encontramos ante un auténtico hito en la historia de la computación, o simplemente una tendencia pasajera que vaya a ser sustituida por algún tipo de evolución en el medio plazo.

8.1.2. Composición y Disposición de los Grafos de Twitter

En cuanto al segundo bloque de las conclusiones, relativo a la composición y disposición de las sociedades formadas a partir de los servicios de redes sociales, se han observado ciertos paralelismos de éstas con las sociedades corrientes, todos ellos relacionados con el reparto de los recursos y la repercusión de los mensajes de sus miembros. Las similitudes encontradas en ambos entornos son las siguientes:

- **La distribución de los recursos entre la población es muy desigual,** una minoría de individuos posee la mayoría de los recursos mientras que la pequeña parte restante está repartida entre muchos otros. Tomando el número de relaciones de cada usuario como los recursos dentro del contexto del servicio de red social *Twitter*, se ha observado una distribución muy similar a las que se dan en las sociedades corrientes. De igual forma que en el mundo real los mayores niveles de riqueza están en manos de una minoría muy reducida, en *Twitter* un pequeño porcentaje de usuarios aglutina la gran mayoría de las relaciones entre ellos.
- **El nivel de recursos actual es determinante en la obtención de nuevos recursos.** El principio utilizado en el contexto económico de la distribución de la riqueza conocido como "the rich get richer", también es aplicable en *Twitter*. Los usuarios con gran número de relaciones poseen mayor facilidad para obtener otras nuevas, al igual que en las sociedades reales los individuos con grandes riquezas tienen una elevada predisposición a aumentarlas o en todo caso a mantenerlas.
- **La repercusión de los mensajes de un usuario o individuo está directamente relacionado con el volumen de su audiencia.** Tal y como ocurre en nuestras sociedades la repercusión que puede tomar un mensaje no depende tanto de su contenido y de su naturaleza, si no del volumen de su audiencia.

Todas estas conclusiones vienen a confirmar la hipótesis de que las sociedades corrientes y las sociedades paralelas que proliferan a partir de los servicios de redes sociales comparten importantes similitudes, pues el comportamiento de una persona en el mundo real no difiere en exceso a su comportamiento como usuario a través de un servicio de red social.

Por otra parte sobre la estructura de interna de *Twitter*, debe decirse que se trata de un servicio de red social caracterizado por el gran número de usuarios que lo componen (a pesar de que la compañía se niega a ofrecer datos, algunas fuentes[83] los cifran en más de 100 millones), pero no por el grado de interconexión existente entre ellos. Ello se debe principalmente a los siguientes motivos:

- **Existe una fuerte dicotomía en la clasificación de sus usuarios**. Como se ha demostrado en el subapartado 6.2.2, la mayoría de los usuarios del servicio de red social pueden clasificarse en dos únicos grupos: uno minoritario que posee un muy abundante número de relaciones, y otro mayoritario apenas conectado con el resto de la red.
- **Ha experimentado un vertiginoso crecimiento en su número de usuarios**. Dicho crecimiento se ha producido un ritmo muy veloz[84], sin ir acompañado por un desarrollo de las conexiones o relaciones de los nuevos usuarios de la misma magnitud. Como consecuencia, el grado de interconexión o número de conexiones por nodo se ha mantenido bajo, tal y como confirman los resultados obtenidos en el apartado 6.2.3.
- **Posee un elevado porcentaje de usuarios inactivos**. A pesar del ingente número de usuarios que agrupa *Twitter*, muchos estudios realizados sobre el servicio de red social por diversas fuentes[84][85][86] han demostrado que un alto porcentaje de dichos usuarios se encuentra inactivo, es decir, sin interés por enriquecer sus conexiones o actualizar sus “tweets”. Esto

conduce al desagrupamiento de la red en términos generales, con numerosos nodos aislados o pobremente conectados.

- **No proporciona ningún servicio de búsqueda de relaciones potenciales**. Al contrario que otros servicios de redes sociales, *Twitter* no dispone de ningún mecanismo de ese tipo, que podría contribuir al enriquecimiento de las conexiones internas entre nodos dentro de la red.

Como conclusión, por tanto, puede extraerse que las relaciones o conexiones internas entre los usuarios o nodos de *Twitter*, al menos dentro de la muestra del estudio, se encuentran escasamente desarrolladas, muy lejos de su auténtico potencial. Este hecho es fundamental para cualquier tipo de red social, aún más si cabe en *Twitter* por su forma de notificación de mensajes, y constituye en la actualidad su verdadera asignatura pendiente. Sin embargo, a favor del servicio de red social ha de reconocerse que durante los últimos tiempos parece haber tomado conciencia del problema, y su equipo de desarrolladores acaba de lanzar un servicio de búsqueda de relaciones potenciales, conocido como *Who to Follow?*[87].

8.2. Conclusiones Personales

De igual forma que durante de la realización del proyecto se han extraído una serie de conclusiones generales relacionadas con el uso de las plataformas "cloud computing" y de la composición de las redes sociales, dentro del ámbito más personal del autor también se han obtenido otras conclusiones de importancia y utilidad.

La realización del proyecto de fin de carrera supone un hito muy importante en la carrera de todo Ingeniero. Se trata de condensar una gran cantidad de conocimientos asimilados, una serie de hábitos adquiridos y muchos años de esfuerzo en un documento que cierra una etapa, y pasa a convertirse en una pequeña y humilde, pero también firme y legítima, contribución al patrimonio intelectual de la universidad. Un ingeniero no culmina su formación hasta la finalización de su proyecto, pero una vez lo

ha hecho este le sirve para reafirmarse en su posición y sentirse con la valía suficiente de terminar un ciclo y enfrentarse a nuevos retos.

Por otra parte, en el desarrollo del proyecto se han extraído una serie de conclusiones que, si bien no son aplicables a éste proyecto en particular puesto que no ha planteado problemas de derechos de autor con la universidad, ni ha supuesto el desaprovechamiento un potencial uso empresarial, sí hacen referencia a la figura del proyecto de fin de carrera en general. Son por lo tanto, unas reflexiones no centradas en este proyecto de fin de carrera concreto, pero sí extensibles a una gran mayoría de ellos.

En el contexto de los proyectos de fin de carrera, el tema de su propiedad intelectual por parte la universidad genera cierto debate y controversia entre alumnos, tutores y autoridades. Como es sabido todo proyecto de fin de carrera concluido pasa a formar parte del patrimonio intelectual de la universidad y por lo tanto es esta última la única responsable y beneficiará del mismo. Esto provoca que a menudo el alcance de los proyectos sea limitado y sus objetivos reducidos, pues en muchos casos alumnos y tutores no se sienten adecuadamente motivados por la falta de incentivos no puramente académicos. Este hecho contrasta con un discurso que se ha establecido en nuestra sociedad, más si cabe en los actuales tiempos de crisis, sobre nuestra falta de espíritu emprendedor e interés por la investigación y la innovación. Si ciertamente se trata de una de las principales debilidades de nuestra economía que debe ser fortalecida, pocos entornos como la universidad y sus departamentos de investigación cuentan con la misma capacidad para servir de vivero de nuevos proyectos y fomento del espíritu emprendedor de sus profesores y alumnos.

El principal motivo de que nuestras universidades no actúen como auténticas "cunas de ideas" se debe a la falta de interés y financiación por parte de entidades externas, que provoca que un número de proyectos tengan una naturaleza completa-

mente académica y alejada de sus aplicaciones prácticas e incluso comerciales. Sin embargo, las empresas públicas y privadas ajenas a la universidad han de caer en la cuenta, de que mediante la financiación de proyectos universitarios no solo contribuirán a la loable causa del apoyo a la educación, sino que además podrían obtener interesantes beneficios de los resultados de sus investigaciones o concepción de nuevas ideas. Con un modelo en el que entre universidad y empresa existieran vínculos más estrechos, ambas conseguirían salir fuertemente reforzadas. La universidad podría obtener financiación para proyectos más ambiciosos, por ejemplo mediante cátedras vinculadas a sociedades o compañías, mientras que las empresas podrían tener a su disposición potentes departamentos de I+D+i y participar en la concepción de ideas de negocio que pudieran aportarle beneficios económicos en el futuro.

En definitiva, como conclusión personal se pretenden lanzar una serie de preguntas al lector para que el mismo pueda valorar lo que aquí se propone. ¿Qué tipo de resultados podrían obtenerse de proyectos en los que alumnos y tutores se sintieran más cómodos y motivados? ¿De dónde surgieron las grandes ideas que han impulsado a las empresas tecnológicas que hoy lideran el sector? ¿Por qué grandes proyectos desarrollados en la universidad con enorme potencial práctico han de terminar en un cajón?

9. Trabajos Futuros

En todo proyecto existe un grado de alcance de las conclusiones, que viene determinado por ciertos condicionantes o limitaciones durante su desarrollo, y deja la puerta abierta a posibles trabajos futuros. De igual forma, a menudo surgen con posterioridad nuevas herramientas o campos de aplicación, que podrían haberse tenido en cuenta durante la realización de un proyecto ya completado.

En el caso particular de *Extracción y análisis de información del servicio de red social Twitter, a través de la plataforma de "cloud computing" Google App Engine*, puede hacerse referencia a dos campos de interés para posibles trabajos futuros. El primero de ellos es el empleo de recursos no gratuitos de *GAE,* habilitando una cuenta de facturación para ampliar la muestra de análisis, y el segundo es la extensión del estudio a otros servicios de redes sociales, mediante un mecanismo análogo, para poder comparar resultados, analizando similitudes y diferencias.

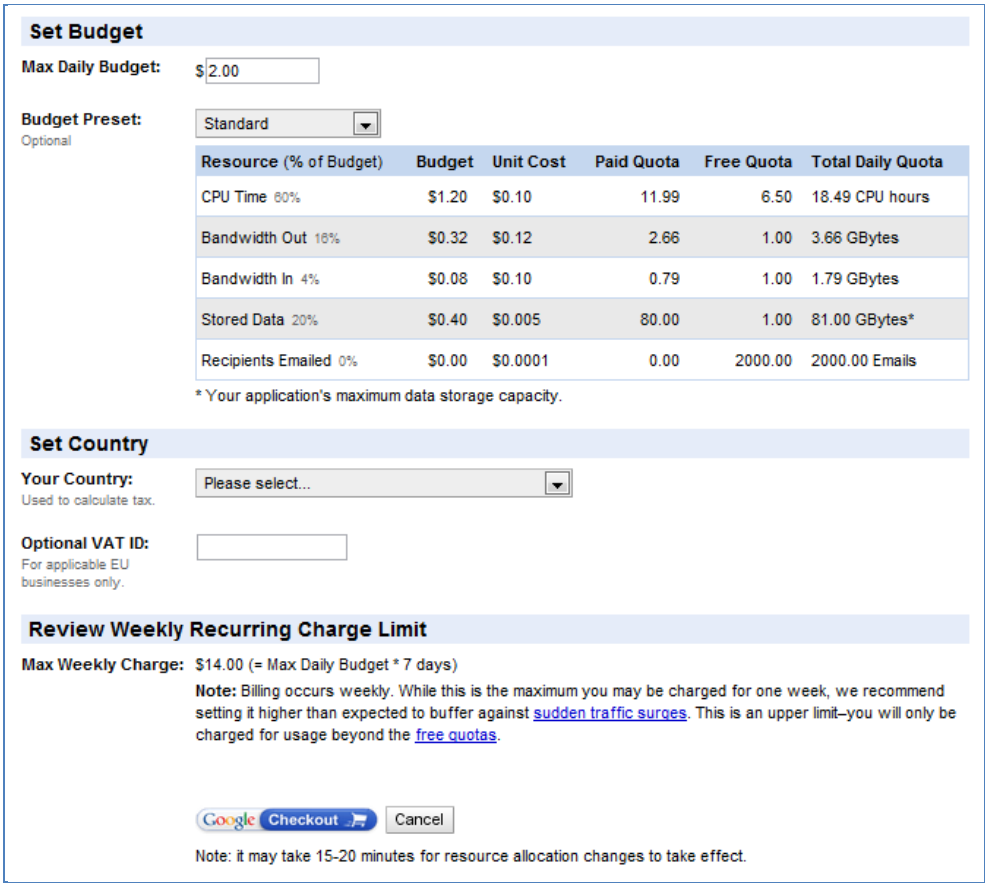
9.1. Empleo de Recursos no Gratuitos de GAE

Uno de los mayores condicionantes del proyecto, si no el mayor, ha sido el no poder plantear el uso de recursos no gratuitos de *Google App Engine* mediante una cuenta de facturación. Este hecho se explica por tratarse de un proyecto universitario de fin carrera, al que no es posible asociar costes ajenos a las horas de trabajo del autor, su tutor y su co-director. En un contexto diferente, como podría ser un proyecto de investigación privado o público, pero ligado a una fuente de financiación externa, el uso de recursos no gratuitos de *GAE* permitiría ampliar en profundidad el estudio con un coste relativamente pequeño.

El empleo de recursos no gratuitos de *Google App Engine* requiere únicamente activar una cuenta de facturación asociada a una forma de pago, y completar un for-

mulario de petición con los recursos deseados a través de la página web habilitada para tal efecto. De forma predeterminada, *GAE* dispone de cuatro configuraciones para la distribución de los recursos adquiridos: estándar, uso intensivo de CPU, uso intensivo de ancho de banda y uso intensivo de espacio de almacenamiento. Sin embargo, el usuario también disfruta de la posibilidad de repartir libremente sus fondos entre todos los recursos ofrecidos. Como medida de seguridad, y para evitar consumos indeseados, los pagos se realizan por adelantado, y la aplicación retoma los límites de las cuentas gratuitas una vez los fondos se hayan agotado.

En la siguiente captura, puede contemplarse el formulario para la activación de una cuenta de facturación por un importe determinado:



Captura 10: Formulario de activación de las cuentas de facturación de GAE

Siguiendo el procedimiento descrito, una aplicación dada podría sobrepasar los límites gratuitos que establece *GAE* para ciertos recursos. En concreto, los recursos que han mostrado un mayor nivel de criticidad durante el proceso de extracción han sido el tiempo de CPU y el espacio de almacenamiento, y de nuevo el tiempo de CPU en la fase de recuperación de los datos (véase los subapartados 12.1 y 12.2 respectivamente). En caso de poder relajarse las limitaciones de ambos recursos, sería viable realizar una ampliación del estudio actual con un espectro de muestra de usuarios mucho más amplio, y obtener nuevas conclusiones con un mayor nivel de detalle. Por ejemplo, con un presupuesto total no superior a los 100$ para un calendario de extracción de 2 semanas de duración, como el empleado en el proyecto actual, podría multiplicarse por diez el número de usuarios contenidos en la muestra del estudio de los grafos de amigos y de seguidores. En tal caso, se trataría de un estudio del comportamiento de más de 3.000.000 de usuarios del servicio de red social *Twitter*, lo que permitiría extraer conclusiones aún más precisas y con un mayor nivel de detalle.

9.2. Estudio de Otras Redes Sociales

Desde un primer momento el proyecto se concibió como la extracción masiva de información de un servicio de red social indeterminado, mediante una plataforma alternativa como la herramienta de computación distribuida *Google App Engine*, para su posterior análisis e interpretación. Por lo tanto, el servicio de red social a examinar no se precisó inicialmente, sino que se llevó a cabo un estudio comparativo de varios de ellos con objeto de escoger el más adecuado para dicho proceso. Finalmente se optó por *Twitter*, tal y como se desarrolla y justifica en el subapartado 1.1.1. No obstante, un posible trabajo futuro como extensión del proyecto actual podría significar realizar estudios similares de otros servicios de redes sociales. De esta forma sería posible comparar resultados, analizar similitudes y diferencias, e incluso extraer conclusiones comunes.

Indudablemente, existen ciertos requisitos que un servicio de red social ha de cumplir para poder ser objeto de un estudio comparable al llevado a cabo durante este proyecto: existencia de API para realizar las extracciones automáticas de información, acceso libre a dicha información o al menos cierto grado de accesibilidad, popularidad de la red social, utilidad de la información que presenta, etc. Si bien pocos servicios de red cumplen estos requisitos de la misma forma que lo hace *Twitter*, existen otros como *Facebook*, *LinkedIn* o *Tuenti* con características similares. Los tres disponen de API (ya sea oficial o no), permiten el acceso a una cantidad de información provechosa suficientemente abundante para el estudio, y por último son muy populares. Por consiguiente, podrían ser estudiados mediante un proceso análogo al empleado en el proyecto sin que ello implicara una excesiva carga de trabajo adicional.

Figura 23: Estudio de otras redes sociales

En particular, cada uno de ellos presenta ciertas características especiales:

- ***Facebook***: posee una API[88] muy potente tecnológicamente, aunque algo restrictiva en cuanto a la búsqueda y extracción de la información. Como punto positivo, se trata del servicio de red social más extendido[28] en todo el mundo con un enorme número de usuarios.

- ***LinkedIn***: su API[89] se encuentra abundantemente documentada, pero no presenta excesivas funcionalidades, y su sistema de relaciones no públicas complica la extracción de información.
- ***Tuenti***: aunque su API oficial no sea pública, existen infinidad de APIs no oficiales desarrolladas que podrían permitir extraer información del servicio de red social mayoritario en España.

Como ya se ha mencionado anteriormente, el estudio de otras redes sociales sería útil para poder comparar resultados, analizando similitudes y diferencias. En concreto, podría ser especialmente interesante verificar si existe algún tipo de diferencia en el comportamiento entre los usuarios que pertenecen a una red de contactos profesionales, como *LinkedIn*, y los que corresponden a redes relacionadas con la amistad y el ocio. Otra posibilidad sería contrastar los datos obtenidos de los usuarios españoles, por ejemplo los usuarios del servicio de red social *Tuenti*, con los de los usuarios internacionales de otras redes como *Twitter* o *Facebook*.

10. Agradecimientos

Este apartado ha sido especialmente reservado para expresar mi agradecimiento a todas aquellas personas que con su apoyo, enseñanzas, consejos y ánimos me han ayudado enormemente en la realización de este proyecto de fin de carrera.

Mi primera mención ha de ser necesariamente para mis padres, hermanos y amigos, sin los cuales no habría podido completar ni éste ni otros muchos hitos importantes de mi vida personal, académica y profesional. Su motivación fue clave durante todo el proceso, y soy consciente de que todos ellos creyeron y confiaron en mí tanto en los buenos como en los malos momentos, lo que reafirmó el vigor de su apoyo.

A continuación quiero referirme al buen y constante trabajo de mi tutor del proyecto, *Juan Manuel Tirado*, y de su co-director, *Daniel Higuero*. Ambos me guiaron con sabios consejos y recomendaciones, y supieron entender mi complicada situación a la hora de compatibilizar el trabajo y la finalización del proyecto durante parte del proceso. De igual forma, me gustaría agradecer al profesor *Florin Isaila* de la *Escuela Politécnica Superior de la Universidad Carlos III de Madrid*, haberme puesto en contacto con ellos y sus clases de *Computer Architecture II*, que me ayudaron a comprender el verdadero potencial de los sistemas distribuidos.

Aunque no de forma tan directa como las anteriores, infinidad de personas y organizaciones también me prestaron apoyo con sus conocimientos y consejos a través de Internet, mediante tutoriales, foros, aplicaciones, etc. De este numeroso y variado grupo, cabe destacar entre otros al desarrollador de *Twitter4J yusukey*, a todo el equipo de personas que se encuentra detrás de la herramienta *Graph-Tool* y a los empleados de *Google* encargados del desarrollo de la plataforma *GAE* y la redacción de sus concisos y útiles tutoriales.

Por último y de forma general, me gustaría también dar las gracias a todos los compañeros, profesores, tutores y demás personas con las que en algún momento coincidí en mi etapa universitaria, por todas aquellas experiencias que compartimos juntos y sin duda contribuyeron, de una u otra forma, a la consecución del objetivo que alcanzo con la presentación de este proyecto de fin de carrera: obtener la titulación como Ingeniero Superior en Informática.

11. Bibliografía y Referencias

1. **Gartner, Inc.** Gartner Identifies the Top 10 Strategic Technologies for 2010. [En línea] http://www.gartner.com/it/page.jsp?id=1210613.

2. **IT Strategists, Inc.** Top Technology Trends for Businesses. [En línea] http://www.itstrategists.com/Technology-Trends.aspx.

3. **Telecom News.** Top 10 Hot Technology Trends for 2010. [En línea] http://vartips.com/carriers/verizon-business/top-10-hot-technology-trends-for-2010-760.html.

4. **Twitter, Inc.** Portal de Twitter. [En línea] http://twitter.com.

5. **Google, Inc.** Google App Engine (GAE). [En línea] http://code.google.com/intl/es-ES/appengine.

6. **Alexa.** The Top 500 Global Sites. [En línea] http://www.alexa.com/topsites.

7. **Google, Inc.** Portal de Google. [En línea] http://www.google.com.

8. **Baidu.com, Inc.** Portal de Baidu.com. [En línea] http://www.baidu.com.

9. **Facebook, Inc.** Portal de Facebook. [En línea] http://www.facebook.com.

10. **Wikimedia Foundation, Inc.** Wikipedia. [En línea] http://www.wikipedia.org.

11. **YouTube, LLC (Google, Inc.).** Portal de Youtube. [En línea] http://www.youtube.com.

12. **Google, Inc.** Portal de Blogger.com. [En línea] https://www.blogger.com.

13. **Yahoo!, Inc.** Portal de Yahoo! [En línea] http://www.yahoo.com.

14. **Microsoft Corporation.** Portal de Windows Live. [En línea] http://www.live.com.

15. —. Portal de MSN. [En línea] http://www.msn.com.

16. **TechCrunch.** *Costolo: Twitter Now Has 190 Million Users Tweeting 65 Million Times A Day.* [En línea] Junio de 2010. http://techcrunch.com/2010/06/08/twitter-190-million-users/.

17. **Bernardo Hernández.** ¿Qué es el 'cloud computing'? *CincoDías.* 1 de 2008, http://www.cincodias.com/articulo/empresas/cloud-computing/20080104cdscdiemp_21/.

18. *How Cloud Computing Is Changing the World.* **Rachael King.** http://www.businessweek.com/technology/content/aug2008/tc2008082_445669.htm , 8 de 2008, Bloomberg Businessweek.

19. **Dans, Enrique.** *El Blog de Enrique Dans.* [En línea] 8 de 2008. http://www.enriquedans.com/2008/08/cloud-computing-por-todas-partes.html.

20. **Cloud Computing (Google Groups).** *List of Cloud Platforms, Providers, and Enablers.* [En línea] http://groups.google.com/group/cloud-computing/web/list-of-cloud-platforms-providers-and-enablers?pli=1.

21. **Amazon.com, Inc.** Amazon EC2. [En línea] http://aws.amazon.com/ec2.

22. **Microsoft Corporation.** Windows Azure. [En línea] http://www.microsoft.com/windowsazure.

23. —. *Windows Azure Platform Introductory Special.* [En línea] http://www.microsoft.com/windowsazure/offers/popup/popup.aspx?lang=en&locale=en-US&offer=MS-AZR-0001P.

24. **Linkedln.** Portal de Linkedln. [En línea] http://www.linkedin.com.

25. **Last.fm Ltd.** *Last.fm.* [En línea] http://www.lastfm.com/.

26. **Comofuncionatwitter.com.** En 2009 Facebook llegó a 350 millones de usuarios. [En línea] http://www.comofuncionatwitter.com/blog/en-2009-facebook-llego-a-350-millones-de-usuarios.

27. **Hopkins, Jeanne.** Facebook Growth Jumps to 350 million Users in 2009. *Hubspot.* [En línea] http://blog.hubspot.com/blog/tabid/6307/bid/5479/Facebook-Growth-Jumps-to-350-million-Users-in-2009.aspx.

28. **Ebizmba.** The 20 Most Popular Social Networking Websites February 2010. [En línea] http://www.ebizmba.com/articles/social-networking-websites.

29. **MySpace.** Portal de MySpace. [En línea] http://www.myspace.com.

30. **Classmates Online, Inc.** Portal de Classmates. [En línea] http://www.classmates.com.

31. **Ning, Inc.** Portal de Ning. [En línea] http://www.ning.com.

32. **AOL, Inc.** Portal de Bebo. [En línea] http://www.bebo.com.

33. **Hi5 Networks.** Portal de Hi5. [En línea] http://www.hi5.com.

34. **Tagged, Inc.** Portal de Tagged. [En línea] http://www.tagged.com.

35. **MyYearbook.com.** Portal de MyYearbook. [En línea] http://www.myyearbook.com.

36. **Tuenti.** Portal de Tuenti. [En línea] http://www.tuenti.com.

37. **Yahoo!, Inc.** Portal de Flickr. [En línea] http://www.flickr.com.

38. **CalcularPageRank.net.** Twitter.com tiene PageRank 9. [En línea] http://www.calcularpagerank.net/twitter.com.

39. **The Nielsen Company.** Twitter Grows 1,444% Over Last Year; Time on Site Up 175%. [En línea] http://blog.nielsen.com/nielsenwire/nielsen-news/twitter-grows-1444-over-last-year-time-on-site-up-175.

40. **Morozov, Evgeny.** Iran Elections: A Twitter Revolution? *The Washington Post.* Online, 2009, Miercoles, 17 Junio.

41. **Twitter, Inc.** Perfil de Twitter de BarackObama. [En línea] http://twitter.com/BarackObama.

42. —. Perfil de Twitter de ladygaga. [En línea] http://twitter.com/ladygaga.

43. —. API de Twitter. [En línea] http://apiwiki.twitter.com.

44. —. Perfil de Twitter de twitterapi. *Perfil Twitter de twitterapi.* [En línea] http://twitter.com/twitterapi.

45. *Cloud Computing and Grid Computing 360-Degree Compared.* **Foster, Ian, y otros.** 2008, Grid Computing Environments Workshop.

46. **Google, Inc.** Google Corporate. [En línea] http://www.google.com/corporate.

47. **Python Software Foundation.** Python. [En línea] http://www.python.org.

48. **Sun Microsystems, Inc. (Oracle Corporation).** Java. [En línea] http://java.com.

49. **Wikipedia.** Small World Experiment. [En línea] http://en.wikipedia.org/wiki/Small_world_experiment.

50. **Columbia University.** Small World Project. [En línea] http://smallworld.columbia.edu.

51. **Watts, Duncan J.** *Small Worlds: The Dynamics of Networks between Order and Randomness.* Priceton, New Jersey : Princeton University Press, 1999. ISBN: 0-691-00541-9.

52. **Twitter4J.** Portal de Twitter4J. [En línea] http://twitter4j.org.

53. **Google, Inc.** GQL Reference. [En línea] http://code.google.com/intl/es/appengine/docs/python/datastore/gqlreference.html.

54. **The Eclipse Foundation.** Eclipse. [En línea] http://www.eclipse.org.

55. **Java-Twitter.** Portal de Java-Twitter. [En línea] http://code.google.com/p/java-twitter.

56. **JTwitter.** Portal de JTwitter. [En línea] http://www.winterwell.com/software/jtwitter.php.

57. **TweetStream4J.** Portal de TweetStream4J. [En línea] http://github.com/seejohnrun/tweetStream4J.

58. **Twitter4J.** Twitter4J Mailing List. [En línea] http://groups.google.com/group/twitter4j.

59. **Twitter, Inc.** Perfil Twitter de yusukey. *Perfil de Twitter de yusukey.* [En línea] http://twitter.com/yusukey.

60. **Universidad de California, Berkeley.** Licencia BSD. [En línea] http://www.bsd.org/.

61. **Oracle Corporation.** MySQL. [En línea] http://www.mysql.com.

62. **GNU Project.** GNU GPL. [En línea] http://www.gnu.org/licenses/licenses.es.html.

63. **Projects Forked.** Graph-Tool. [En línea] http://projects.forked.de/graph-tool/.

64. **SciPy Organization.** PyLab. [En línea] http://www.scipy.org/PyLab.

65. **Twitter, Inc.** *Perfil de Twitter de guillermocbns.* [En línea] http://twitter.com/guillermocbns.

66. **World Wide Web Consortium.** Hypertext Transfer Protocol HTTP/1.1. [En línea] http://www.w3.org/Protocols/rfc2616/rfc2616.html.

67. **Twitter, Inc.** *Perfil de Twitter de zaquito.* [En línea] http://twitter.com/zaquito.

68. **Twitter, Inc.** *Perfil de Twitter de googlappengine.* [En línea] http://twitter.com/googlappengine.

69. *Measuring User Influence in Twitter: The Million Follower Fallacy.* **Haddad, Hamed, y otros.** Washington : Proc. International AAAI Conference on Weblogs and Social Media (ICWSM), 2010.

70. **Wikipedia.** *Power law.* [En línea] http://en.wikipedia.org/wiki/Power_law.

71. —. Vilfredo Federico Damaso Pareto. [En línea] http://es.wikipedia.org/wiki/Vilfredo_Pareto.

72. —. Joseph Juran. [En línea] http://es.wikipedia.org/wiki/Joseph_Juran.

73. —. Principio de Pareto. [En línea] http://es.wikipedia.org/wiki/Principio_de_Pareto.

74. —. Distribución de Pareto. [En línea] http://es.wikipedia.org/wiki/Distribución_de_Pareto.

75. **Shirky, Clay.** *Power Laws, Weblogs, and Inequality.* [En línea] February de 2003 . http://www.shirky.com/writings/powerlaw_weblog.html.

76. **Stephen, Andrew T. y Toubia, Olivier.** Explaining the Power-Law Degree Distribution in a Social Commerce Network. *Social Networks.* Columbia : Working Paper Series, 2009, pp. 262-270.

77. **Adamic, Lada A., y otros.** *Search in power-law networks.* [En línea] 2001. http://www.hpl.hp.com/research/idl/papers/plsearch/pre46135.pdf.

78. **Facebook Blog.** Friend Suggestor. [En línea] http://blog.facebook.com/blog.php?post=18829762130.

79. **Hofman, Jake.** *Social network analysis with Hadoop.* [En línea] Octubre de 2009. http://jakehofman.com/talks/hadoopworld_20091002.pdf.

80. **Java, Akshay, y otros.** *Why We Twitter: Understanding Microblogging.* [En línea] Septiembre de 2007. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.87.6790&rep=rep1&type=pdf.

81. **Escuela Politécnica Superior de la Universidad Carlos III de Madrid.** http://www.uc3m.es/portal/page/portal/administracion_campus_leganes_est_cg/proyecto_fin_carrera#topofmypage. *Formulario_PresupuestoPFC-TFG.xlsx.* [En línea]

82. **Google, Inc.** *Facturación y presupuestado de recursos.* [En línea] http://code.google.com/intl/es-ES/appengine/docs/billing.html.

83. **The Huffington Post.** Twitter User Statistics REVEALED. *The Huffington Post.* Online, 2010, Domingo, 4 de Abril.

84. **Dan Zarrella.** State of the Twittersphere June 2009. *Hubspot.* [En línea] Junio de 2009. http://blog.hubspot.com/Portals/249/sotwitter09.pdf.

85. **Barracuda Labs.** Annual Report 2009 Barracuda Labs. [En línea] Enero de 2010. http://barracudalabs.com/downloads/BarracudaLabs2009AnnualReport-FINAL.pdf.

86. **The Metric System .** New Data on Twitter's Users and Engagement. [En línea] Enero de 2010. http://themetricsystem.rjmetrics.com/2010/01/26/new-data-on-twitters-users-and-engagement/.

87. **Twitter Blog.** Discovering Who To Follow. [En línea] Julio de 2010. http://blog.twitter.com/2010/07/discovering-who-to-follow.html.

88. **Facebook Developers.** API oficial de Facebook. [En línea] http://developers.facebook.com/docs.

89. **LinkedIn Developer Network.** API de LinkedIn. [En línea] http://developer.linkedin.com/community/apis.

12. Apéndice

El apéndice del proyecto contiene las gráficas de los recursos consumidos en el entorno "cloud" durante las fases de extracción de los grafos sociales, y de lanzamiento de las peticiones de recuperación. Asimismo, se incluyen ejemplos de capturas de la *Consola de Administración* de *Google App Engine*, realizadas durante el proyecto.

12.1. Recursos "Cloud" durante la Fase de Extracción de los Grafos Sociales

En los siguientes subapartados, se exponen los valores medios de los recursos consumidos en el entorno "cloud" por los extractores de amigos y seguidores, durante la fase de extracción de los grafos sociales.

12.1.1. Extractores de Amigos

El Gráfico 20 muestra el tiempo medio de CPU y el del almacén de datos:

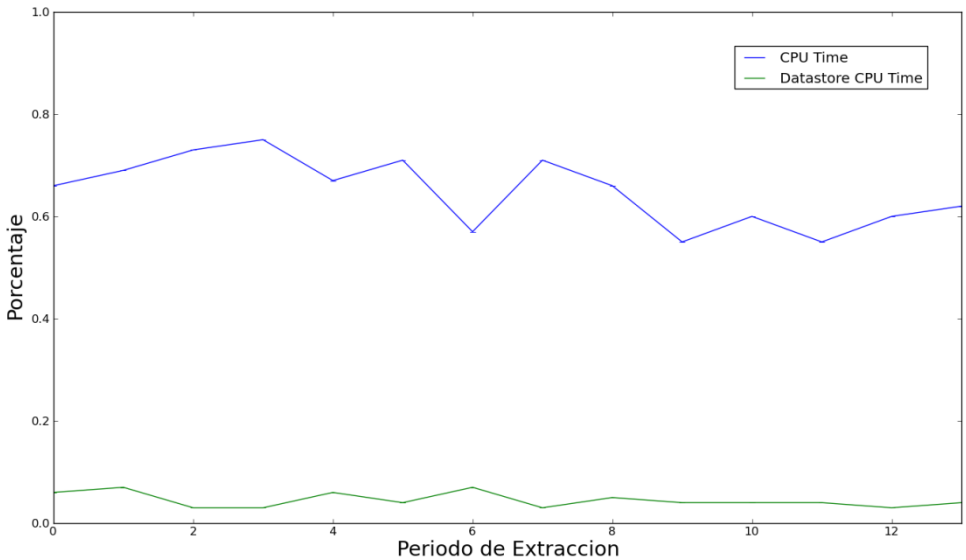


Gráfico 20: Recursos "cloud" de la fase de extracción de amigos

12.1.2. Extractores de Seguidores

El Gráfico 21 muestra el tiempo medio de CPU y el del almacén de datos:

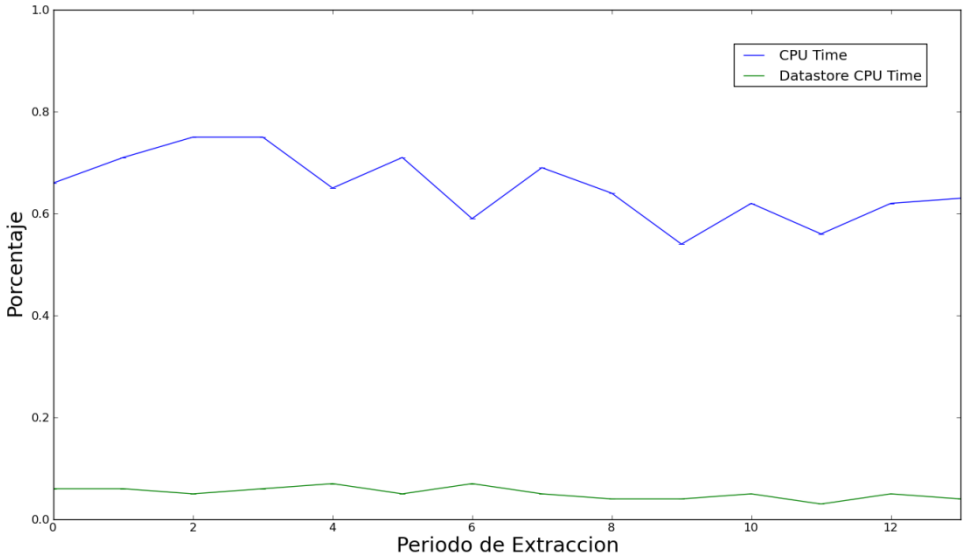


Gráfico 21: Recursos "cloud" de la fase de extracción de seguidores

12.2. Recursos "Cloud" durante la Fase de Lanzamiento de las Peticiones de Recuperación

En los siguientes subapartados se muestran los valores medios de los recursos consumidos en el entorno "cloud" por los extractores de amigos y seguidores, durante la fase de recuperación de los grafos sociales.

12.2.1. Extractores de Amigos

El Gráfico 22 muestra el tiempo medio de CPU y el del almacén de datos:

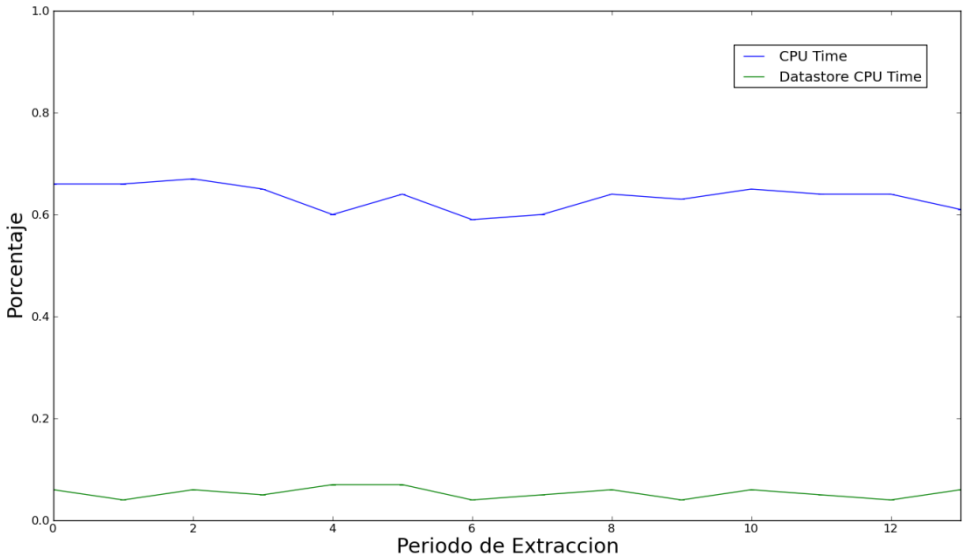


Gráfico 22: Recursos "cloud" de la fase de lanzamiento de peticiones de recuperación de amigos

12.2.2. Extractores de Seguidores

El Gráfico 23 muestra el tiempo medio de CPU y el del almacén de datos:

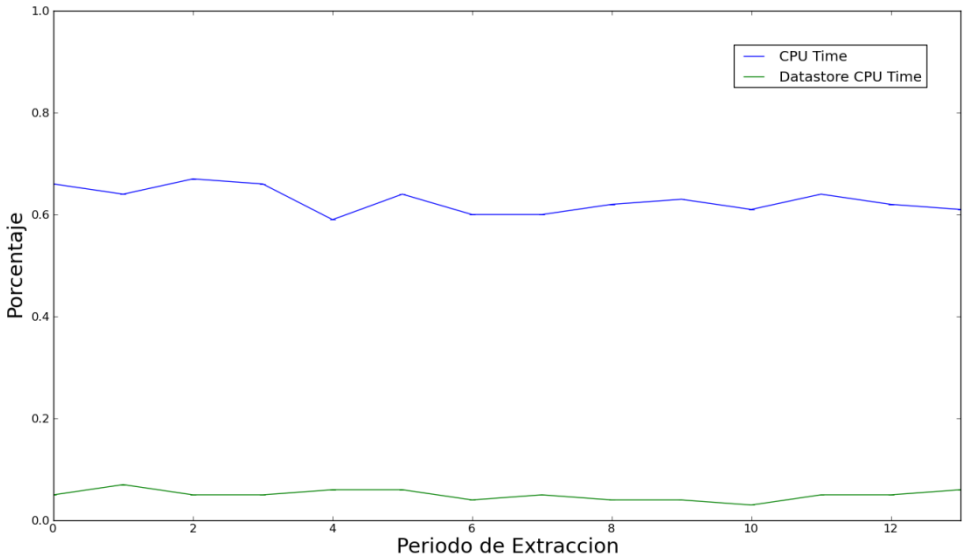


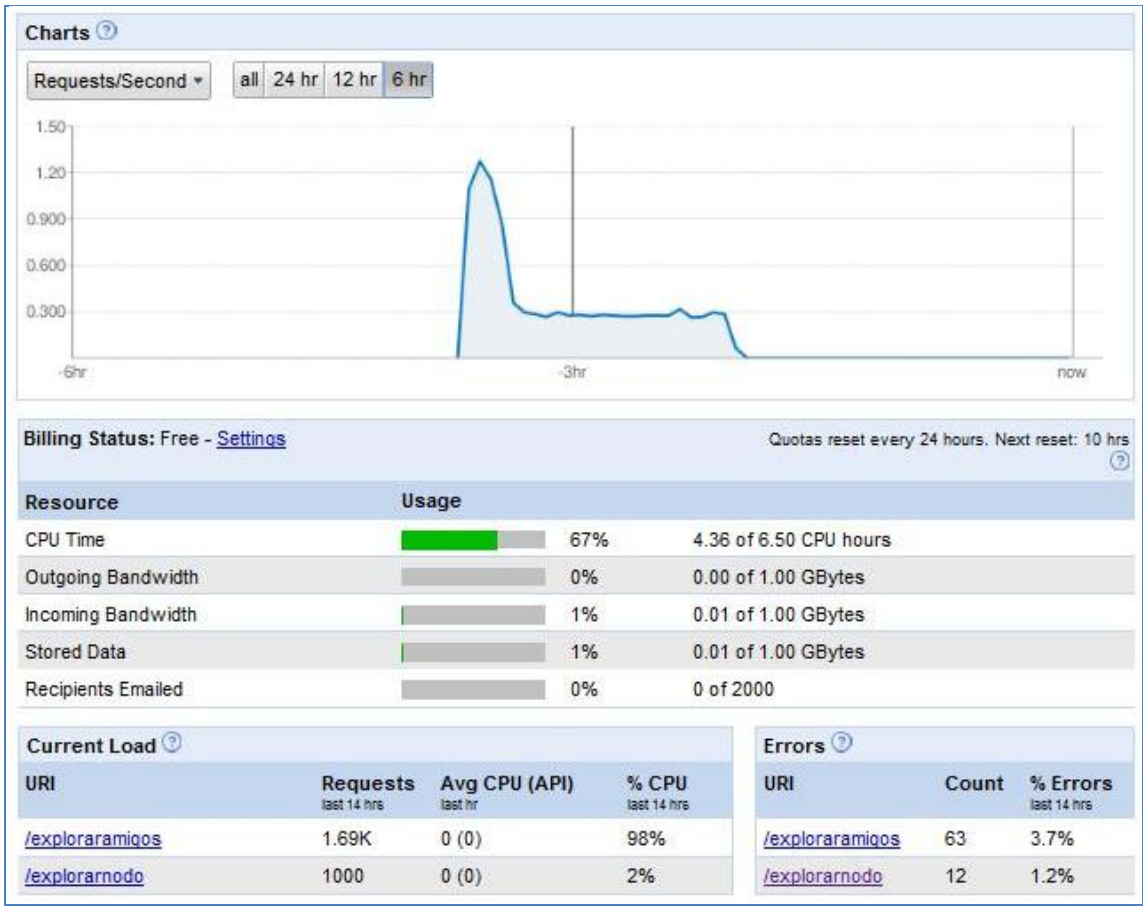
Gráfico 23: Recursos "cloud" de fase de lanzamiento de peticiones de recuperación de seguidores

12.3. Capturas de la Consola de Administración de GAE

En los siguientes subapartados, se muestran algunos ejemplos de capturas de la *Consola de Administración* de *GAE* durante distintas fases del proyecto.

12.3.1. Fase de Extracción de los Grafos Sociales

La siguientes capturas (Captura 11 y Captura 12) muestran los detalles de cuota de recursos *GAE* y de monitorización para el extractor de amigos número uno (nodos desde el 0 hasta el 999), durante el primer periodo de la fase de extracción de los grafos sociales.



Captura 11: Dashboard 1º periodo de extracción, extractor de amigos número 1

Requests Quotas are reset every 24 hours. Next reset: 10 hours

Resource	Daily Quota			Rate
CPU Time		67%	4.36 of 6.50 CPU hours	Okay
Requests		0%	2687 of 1333328	Okay
Outgoing Bandwidth		0%	0.00 of 1.00 GBytes	Okay
Incoming Bandwidth		1%	0.01 of 1.00 GBytes	Okay
Secure Requests		0%	0 of 1333328	Okay
Secure Outgoing Bandwidth		0%	0.00 of 1.00 GBytes	Okay
Secure Incoming Bandwidth		0%	0.00 of 1.00 GBytes	Okay
Datastore				
Datastore API Calls		1%	148327 of 10368000	Okay
Stored Data		1%	0.01 of 1.00 GBytes	Okay
Data Sent to API		0%	0.01 of 12.00 GBytes	Okay
Data Received from API		0%	0.01 of 116.00 GBytes	Okay
Datastore CPU Time		6%	3.85 of 62.11 CPU hours	Okay

Captura 12: Quota details 1º periodo de extracción, extractor de amigos número 1

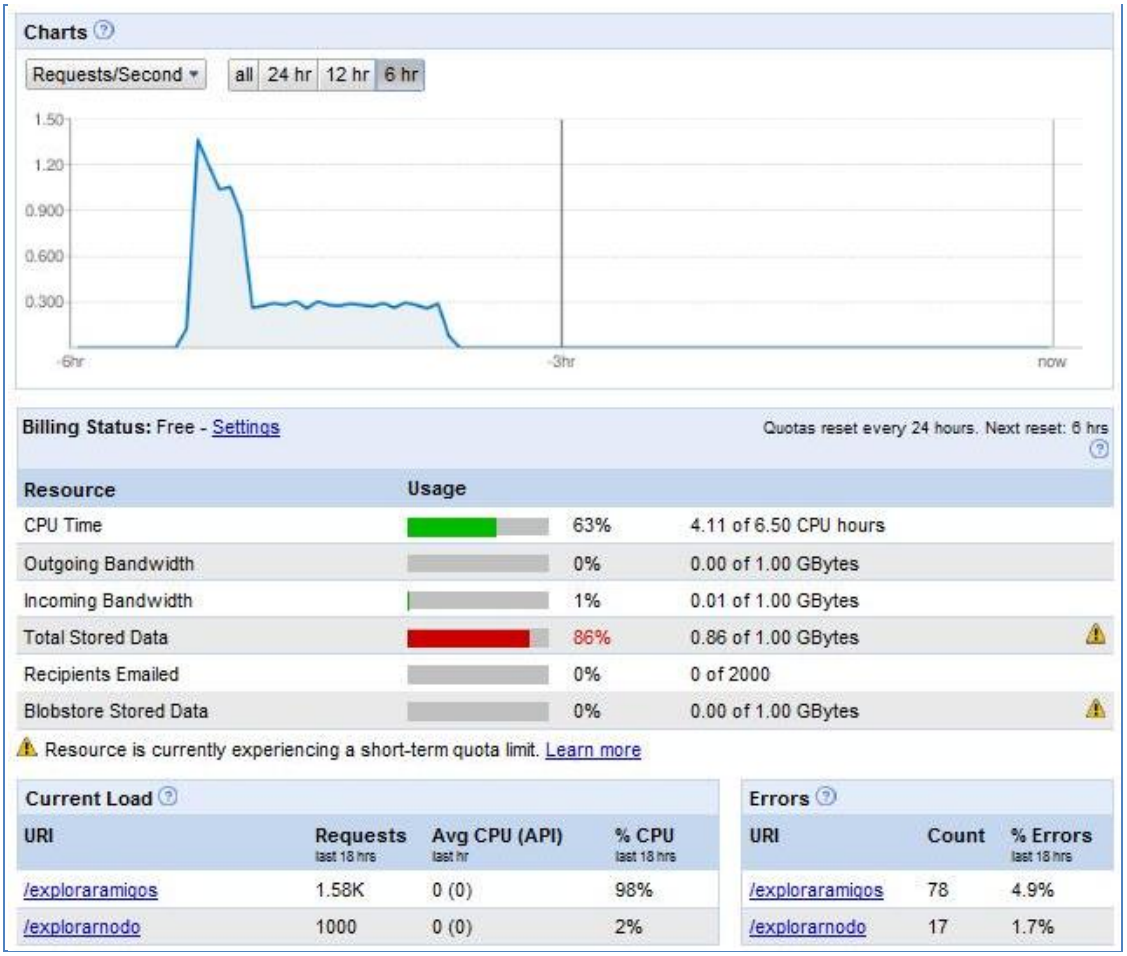
12.3.2. Fase de Lanzamiento de las Peticiones de Recuperación

A continuación se presentan las capturas análogas para la fase de recuperación.

Requests Quotas are reset every 24 hours. Next reset: 6 hours

Resource	Daily Quota			Rate
CPU Time		63%	4.11 of 6.50 CPU hours	Okay
Requests		0%	2582 of 1333328	Okay
Outgoing Bandwidth		0%	0.00 of 1.00 GBytes	Okay
Incoming Bandwidth		1%	0.01 of 1.00 GBytes	Okay
Secure Requests		0%	0 of 1333328	Okay
Secure Outgoing Bandwidth		0%	0.00 of 1.00 GBytes	Okay
Secure Incoming Bandwidth		0%	0.00 of 1.00 GBytes	Okay
Storage				
Datastore API Calls		1%	142957 of 10368000	Okay
Blobstore API Calls		0%	0 of 10368000	Okay
Total Stored Data		86%	0.86 of 1.00 GBytes	Limited
Blobstore Stored Data		0%	0.00 of 1.00 GBytes	Limited
Data Sent to Datastore API		0%	0.01 of 12.00 GBytes	Okay
Data Received from Datastore API		0%	0.01 of 116.00 GBytes	Okay
Datastore CPU Time		6%	3.71 of 62.11 CPU hours	Okay

Captura 13: Quota details 1º periodo de recuperación, extractor de amigos número 1



Captura 14: Dashboard 1º periodo de recuperación, extractor de amigos número 1